

MXK Configuration Guide

For software version 2.5

December, 2014

Document Part Number: 830-01812-22



Z H O N E TM

Zhone Technologies
@Zhone Way
7195 Oakport Street
Oakland, CA 94621
USA
510.777.7000
www.zhone.com
info@zhone.com

COPYRIGHT C2000-2014 Zhone Technologies, Inc. and its licensors. All rights reserved.

This publication is protected by copyright law. No part of this publication may be copied or distributed, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language in any form or by any means, electronic, mechanical, magnetic, manual or otherwise, or disclosed to third parties without the express written permission from Zhone Technologies, Inc.

Bitstorm, EtherXtend, IMACS, MALC, MXK, Raptor, SLMS, Z-Edge, Zhone, ZMS, zNID, MX, MXP and the Zhone logo are trademarks of Zhone Technologies, Inc.

Zhone Technologies makes no representation or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability, non infringement, or fitness for a particular purpose. Further, Zhone Technologies reserves the right to revise this publication and to make changes from time to time in the contents hereof without obligation of Zhone Technologies to notify any person of such revision or changes.

TABLE OF CONTENTS

About This Guide	27
Style and notation conventions	27
Typographical conventions	28
Related documentation	28
Acronyms	29
Contacting Global Service and Support	30
Technical support	30
Hardware repair	31
 Chapter 1	
MXK	33
MXK overview	33
MXK chassis cards	33
MXK uplink cards	34
MXK line cards	35
MXK specifications	39
Management	39
IP and data support	39
Rate Limiting	40
VoIP	40
MGCP	41
SIP	41
 Chapter 2	
MXK Operations, Administration, and Maintenance	43
MXK device management	43
Overview of MXK device management	43
Manage the MXK from the CLI	44
Log into the serial (craft) port	45
Out-of-band management on the MXK	47
In-band management on the MXK	49
Manage the MXK from ZMS	59
Configure the MXK to run ZMS in SNMPv3	59
Mass provisioning from the CLI when running ZMS	60
Manage the MXK from the WebUI	63
Manage the MXK using Zhone Web User Interface	64
Disable the Web UI	65
MXK system administration	67
MXK system defaults	67
Defaults overview	67
Monitoring the MXK through the serial craft port	68
Enable/disable temporary logging sessions	68
User account administration	68
Add users	69
Create an SNMP v3 user from CLI	70
Change default user passwords	70

Delete users	70
Delete the admin user account.....	71
Reset passwords	71
View chassis and system information.....	73
MXK 819 and 823 fan tray monitoring.....	74
MXK 319 fan tray monitoring.....	75
MXK built-in alarm input output	77
View runtime statistics for the MXK with the card stats command.....	79
Monitor the system with log files	81
Overview	82
Default log store level	82
User login notification.....	82
Enable/disable logging	82
Log message format	83
Modify logging levels	84
Non-persistent log messages	86
Persistent log messages	88
Example log messages.....	88
Log filter command.....	88
Send messages to a syslog server.....	89
Specify different log formats for system and syslog messages.....	91
Navigate the MXK file system	93
Access the MXK file system.....	93
Download software files	94
MXK basic system administration commands	96
Commands: new, list, show, get, update, delete.....	96
Commands: interface show, bridge show.....	103
Commands: bridge stats	104
Commands: port show, port up, port down, port bounce, port status	105
Save and restore configurations	105
SNTP.....	106
Set system for SNTP	106
Set Daylight Savings Time begin and end times.....	107
MXK Simple Network Management Protocol (SNMP).....	108
Create SNMP community names and access profiles.....	108
Configure traps.....	109
MXK port management.....	111
Port command overview	111
View the administrative and operational states of ports with the port status and port show command.....	112
port status and port show command	112
View DDM data on Ethernet SFPs with the port show command	112
DDM data on Ethernet SFPs overview	113
DDM data on Ethernet line card Ethernet SFPs.....	113
DDM data on uplink card Ethernet SFPs	114
Change port administrative states with the port testing, up, down, or bounce commands	115
port testing command	115
port up command.....	116

port down command	116
port bounce command	117
Port descriptions on the MXK	117
Port description rules	117
Add, modify, list, and delete a port description	118
Search port descriptions	122
Port mirroring.....	123
port mirror command syntax	123
Create a mirrored port on the uplink card	124
Ethernet Jumbo Frames.....	126
MXK security	128
MXK security (SSH, SFTP, and HTTPS)	128
Enable security on the MXK	128
DSA and RSA keys	130
Tested MXK SSH clients	130
Encryption-key commands	131
Port access security	132
Radius support	134
MXK alarms	139
Alarm manager.....	139
Alarm suppression	140
Configurable high and low chassis temperature alarms	142
MXK card configuration	148
View uplink cards	148
View line cards	148
MXK card configuration.....	149
Add a card profile.....	150
Delete a card profile	151
Add a card that returns parameter prompts	152
card stats command	155
MXK DNS resolver configuration	157
CPE Manager	158
Accessing the CPE's private address, ports.....	159
Viewing the CPE Manager ports	163
Troubleshooting CPE Manager.....	165
Additional information about CPE manager.....	167
Web UI cut-through for EtherXtend devices.....	168
Web UI cut-through for EtherXtend devices.....	170
Chapter 3 MXK Clocking	173
Clock management on the MXK overview1	173
MXK local and system clocking	174
Local clocking source on the MXK	174
System clocking on the MXK overview	174
Set MXK system clocking from MXK sources	177
MXK system clocking	177

system-clock-profile overview.....	177
Configure MXK line and uplink cards for system clocking.....	180
Set a line card as the clocking source.....	180
Set a CLK or TOP uplink card as the clocking source.....	181
Precision Time Protocol (PTP) and SyncE clock management on the MXK	
184	
Ordinary clock and boundary clock PTP configurations.....	184
MXK Ordinary Clock.....	184
MXK Boundary Clock	185
SyncE clock management.....	198
 Chapter 4 MXK Bridge Configuration	201
Overview of bridging on the MXK	201
Bridging fundamentals.....	201
Terminology and concepts.....	203
Physical port	204
Physical interface	204
Logical interface.....	205
Bridges and bridge interfaces.....	205
VLANs and SLANs, untagged, tagged and stagged	205
Upstream and downstream	208
Broadcast, multicast, and unicast	209
Tagging operations.....	209
Tagging operations overview	210
Common tagging operation scenarios	212
MXK bridge types.....	217
Symmetric bridges.....	217
Asymmetric bridges	222
Intralinked bridges.....	226
bridge-path creation with the bridge add command	230
bridge add command	230
bridge add parameters	230
Verify the bridge-interface-record parameters.....	231
Bridge add and bridge-path modify defaults	232
IPv6 compatibility.....	235
Basic bridged data on the MXK	239
Uplink bridges with VLAN ID	239
Downlink bridge-types for asymmetrical bridge configurations	240
downlink-data bridging for data	241
downlink-voice bridging for voice	241
downlink-video bridging for video.....	241
downlink-pppoe bridging for PPPoE	242
downlink-p2p bridging for P2P.....	242
downlink-upmcast bridging for upstream multicast.....	243
user specified bridging	243
Downlink bridges with VLAN ID	243
Untagged downlink bridges on Active Ethernet	244

Tagged downlink bridges on Active Ethernet	245
TLS bridges with VLAN ID	246
TLS bridges	246
TLS bridge parameters floodUnknown and floodMulticast	247
Wire bridge configuration	250
Nonlearning and learning wire bridges	250
GPON wire bridge Q-in-Q-in-Q encapsulation	253
Q-in-Q on bridges (VLAN IDs and SLAN IDs)	254
Overview of Q-in-Q (VLAN/SLAN)	254
Uplink staged bridge to downlink staged bridge	254
Tagged downlink bridge to staged uplink bridge (SLAN promotion)	255
untagged downlink bridge to staged uplink bridge (double-promotion)	257
Delete the uplink and downlink bridges	258
Turn off Q-in-Q for the entire MXK system	258
Q-in-Q-in-Q (VLAN IDs, SLAN IDs and packet rules) on bridges	259
Q-in-Q-in-Q overview	259
Configure an access TLS bridge for Q-in-Q-in-Q	261
Configure a network facing TLS bridge for Q-in-Q-in-Q	262
Bridges using VLAN 0	263
Possible bridging configuration behaviors for VLAN 0	263
Uplink bridges with VLAN 0 SLAN ID staged configuration cases	264
MXK bridging configuration with VLAN 0 on TLS bridges for multi-point connections	266
MXK bridging configuration with VLAN 0 on tagged intralinks	268
MXK bridging configuration with VLAN 0 on staged intralinks	270
Bridges with link aggregation	271
Configure link aggregation uplink bridges	271
Configure link aggregation line card bridges	272
Configure a TLS bridge on a link aggregation bridge	273
Bridge loop prevention	275
Bridge loop prevention overview	276
Configure bridge loop prevention	277
View bridge loop detection alarms	280
View bridge loop prevention on a bridge	281
Unblock the bridge	281
Secure bridging	283
Dynamic IP filtering on a bridge (Secure DHCP)	283
Static IP and MAC for secure bridging on the MXK	284
Broadcast suppression	293
Configure uplink and downlink bridges on GPON for triple-play services	294
Advanced bridged data on the MXK with VLAN translation	298
Overview of VLAN translation on the MXK	298
Possible bridging configuration behaviors for VLAN/SLAN translation	298
bridge show command for VLAN translation	299
Basic VLAN translation on bridges	299
VLAN translation on TLS bridges	299
VLAN translation on asymmetric bridges	301
Advanced VLAN translation on bridges	303
VLAN translation and SLAN promotion on asymmetric bridges	303

Configure asymmetric bridges with SLAN translation (outer tag)	306
Configure asymmetric bridges for VLAN and SLAN translation	308
VLAN translation on Active Ethernet asymmetric bridges with CoS replacement	
311	
Filters for MXK bridges (packet-rule-record)	313
Overview of packet-rule-record filters.....	313
Create packet-rule-record filters.....	314
Packet rule types.....	315
Option 82 DHCP on bridge packet rule (bridgeinsertoption82).....	316
Option 82 for DHCP relay overview.....	317
Option 82 DHCP on bridge packet rule (bridgeinsertoption82) configuration with-	
out macros defined strings	318
Option 82 DHCP on bridge packet rule (bridgeinsertoption82) configuration with	
macro defined strings.....	319
DHCP on bridge packet rules (DHCP relay, and Forbid OUI).....	323
DHCP relay	323
DHCP relay bridge configuration.....	324
Forbid OUI	327
PPPoE with intermediate agent (bridgeinsertpppoevendortag)	327
PPPoE with intermediate agent overview	328
PPPoE with intermediate agent configuration without macro defined strings.....	329
PPPoE with intermediate agent configuration with macro defined strings....	331
Bandwidth limiting by port and service, single and dual rate limiting.....	334
Rate limiting overview	334
Configure color blind rate limiting.....	337
Configure color aware rate limiting	343
Color to Cos default values	347
DSCP to COS (802.1p) mapping	348
Destination MAC swapping.....	352
Bridge storm protection	355
Bridge storm protection overview	355
Default packet rule filters (bridgestormdetect)	355
Case 1: bridgestormdetect packet rule for discard	358
Case 2: bridgestormdetect packet rule for discard + alarm	359
Case 3: bridgestormdetect packet rule for discard + alarm + block.....	360
Modify the default bridgestormdetect rules	361
View detected packets statistics	363
View the packets	364
Unblock a bridge	366
Access Control List (ACL)	367
ACL packet rule filtering rules on the MXK	367
ACL packet rule filtering variables	367
ACL filtering options	368
Configure ACL packet rules.....	370
Additional bridging services	378
PPPoA - PPPoE interworking on bridges	378
Rapid Spanning Tree Protocol (RSTP).....	381
RSTP port role.....	381
RSTP port state.....	382

RSTP on uplinks.....	383
RSTP rlinks	385
Multiple Spanning Tree Protocol (MSTP) on the MXK	389
MSTP overview	390
MSTP instances	390
MSTP port role.....	390
MSTP port states	391
MSTP network routers	393
MSTP network topology planning	393
MSTP network topology components.....	393
MSTP ring configuration.....	395
MSTP ring operation	402
MSTP ring IP on a bridge in-band device management	405
Shaping Traffic: Class of Service Queuing	406
Configuring Class of Service	407
COS and SCOS replacement on Ethernet frames	409
“Denial of Service” prevention	411
Bridging differences between the MALC and MXK.....	412
MXK bridge statistics-on-demand.....	413
Bridge interface statistics-on-demand overview.....	413
bridge statistics commands on bridge interfaces with statistics enabled by default.....	414
View bridge interface statistics that are enabled by default.....	414
Use the bridge stats reset, clear, list, and rules commands for default and enabled statistics.....	415
Bridge statistics-on-demand.....	416
Statistics-on-demand for bridge interface configuration	417
View bridge statistics on Ethernet bridges	417
View bridge statistics on GPON bridges.....	419
Bridge statistics display	423
Administrative commands	424
bridge add/delete commands.....	424
bridge show/showall commands	424
bridge-path add/modify/show/delete commands	425
Chapter 5 Video Configuration	427
MXK bridged video overview.....	427
MXK bridged video with IGMP proxy	428
IGMP proxy overview	428
IGMP proxy join and leave requests.....	428
MXK basic bridged video configuration	429
Basic bridged video with IGMP proxy configuration overview.....	429
Basic video configuration with IGMP proxy.....	429
Advanced bridged video with IGMP and IGMP DSCP configuration.....	433
IGMP DSCP overview.....	433
IGMP DSCP and IGMP with proxy reporting and default IP address.....	435
IGMP DSCP and IGMP with proxy reporting and custom IP address	436

Advanced bridged video on the MXK with VLAN translation and MVR...	439
Bridged video on the MXK with VLAN translation	440
Bridged video on the MXK with MVR	443
Bridged video on the MXK with VLAN translation and MVR.....	447
Bridged video on the MXK with SLAN promotion and MVR	450
Bridged video on the MXK with VLAN translation, SLAN promotion, and MVR	453
Bridged video on the MXK with dual MVR	456
Bridged video with no MVR.....	457
Bridged video with single MVR	457
Bridged video with dual MVR	457
Display bridge IGMP	462
Display bridge IGMP	462
IGMP bridging statistics	463
IGMPv3 and IGMPv2 proxy agent.....	465
IGMPv3	465
IGMPv2	466
 Chapter 6 Voice Configuration.....	467
Voice cards.....	467
VoIP configuration basic steps.....	468
System settings	469
Setting a-law or mu-law and DSP settings	469
Additional system settings	472
Configure an IP interface for voice traffic.....	480
Voice add command	481
SIP	483
SIP server	483
SIP dial plan configuration	485
POTS to VoIP connection with SIP	487
Emergency Stand Alone (ESA) for SIP	489
DSCP marking for SIP and RTP.....	493
Enhanced SIP 911 Service	495
RFC 3262 for SIP	497
SIP PLAR.....	500
SIP PLAR server configuration	500
ESA for SIP PLAR	501
POTS to VoIP connection with SIP PLAR.....	504
ISDN to VoIP connection with SIP PLAR	505
MGCP	507
MGCP server	507
POTS to VoIP connection with MGCP	509
H.248	511
H.248 configuration	511
POTS to VoIP connection with H.248.....	512
ISDN to VoIP connection with H.248	513

ESA for H.248	514
Subscriber voice features configuration	522
Default subscriber voice features	522
Call transfer.....	524
SIP local call conferencing	525
Configuring call conferencing on the MXK.....	525
Connecting three-way conference calls.....	526
Current call conferencing limitations	527
SIP local intercom.....	527
Configuring SIP local intercom feature on the MXK	528
Activating and Deactivating intercom calls	528
Interaction with other features.....	529
Line Side Answer Supervision and reverse battery signal support for payphones	530
DTMF mode support per port basis	533
Data exchange only.....	535
Voice exchange only.....	536
Plar	537
Hotline and Warmline.....	538
Cut-off on Disconnect.....	539
Always off hook.....	540
Centrex	541
Advanced features	542
ESA	542
ToS configuration for voice signaling packet.....	542
T.38 fax	544
T.38 to VoIP connection	544
T.38 fax to Voice Gateway V5.2/GR303 connection with SIP PLAR	547
Route T.38 fax between MXKs with Voice Gateway.....	547
Chapter 7 MXK Pseudo Wire Emulation (PWE) Configuration	549
PWE on the MXK.....	549
Overview.....	550
PWE connections	552
PWE timing	552
The pwe-tdm add command.....	556
PWE IP addresses and UDP ports	557
Channelization: SAToP and CESoP.....	558
Payload size, jitter buffer and filler patterns	559
PWE solution with EAPS	561
Creating PWE connections	562
PWE with T1 or E1.....	562
PWE with CESoP channelization	563
Configuring PWE for E1 ISDN PRI.....	566
Admin up the PWE adminstat and port	568
PWE alarms, logs and traps	570
PWE Loss of Service alarm	570
PWE LOS logs	570

	PWE LOS traps	570
	Troubleshooting LOS	570
	PWE service degradation alarm.....	570
	PWE operational status	573
	PWE commands	576
Chapter 8	Link Aggregation Configuration	589
	Link aggregation overview	589
	Link aggregation and LACP	590
	lacp command	590
	LACP link aggregation active mode.....	591
	Link resiliency	591
	MXK Ethernet ports available for link aggregation	591
	Ethernet uplink ports available for link aggregation	591
	Ethernet uplink card ports available for link aggregation across cards.....	593
	Ethernet line card ports available for link aggregation	593
	Configure link aggregation on Ethernet uplink ports	594
	Configure a Ethernet uplink port for redundant link aggregation.....	594
	Configure multiple Ethernet uplink ports for link aggregation across cards	596
	Delete a link aggregation group	600
	Configure link aggregation on Ethernet line card ports	601
	Configure line card Ethernet ports for LACP	601
	Configure link aggregation bridges	602
	Configure link aggregation uplink bridges	602
	Configure link aggregation line card bridges.....	603
	Configure a TLS bridge on a link aggregation bridge	604
Chapter 9	MXK Ethernet Uplink Cards	607
	MXK 100/1000 Ethernet and 10 GE uplink cards	607
	MXK 100/1000 Ethernet and 10 GE uplink cards overview	608
	MXK Ethernet uplink card specifications.....	609
	MXK uplink card types.....	611
	MXK Ethernet uplink cards with clocking	612
	MXK Ethernet uplink cards with clocking overview	613
	MXK 10-port 2X 10G 8X 1-GE uplink card with Timing over Packet (TOP)	614
	10-port 2X 10G 8X 1-GE uplink card (TOP) overview	614
	MXK-UPLINK-2X10G-8X1G-TOP card specifications.....	615
	MXK 10-port 2X 10G 8X 1-GE uplink card with T1/E1 or BITS timing inputs.	615
	MXK 10-port 2X 10G 8X 1-GE uplink card with T1/E1 or BITS timing inputs	
	overview.....	616
	MXK-UPLINK-2X10G-8X1G-CLK card specifications	617
	MXK 6-port 6X 1-GE uplink card with T1/E1 or BITS timing inputs	617
	MXK 6-port 6X 1-GE uplink card with T1/E1 or BITS timing inputs overview	
	618	
	MXK 6-port 6X 1-GE uplink card with T1/E1 or BITS timing inputs specifications	

619	
MXK uplink cards with clocking card types	619
MXK uplink clocking cards LED redundancy status	620
MXK Ethernet uplink cards pinouts	621
Ethernet port pinouts	621
Clocking port pinouts	622
Serial (craft) port pinouts	622
Cables and clocking	623
Equipment protection and facility protection on the MXK	626
MXK redundant uplinks for equipment protection configuration	626
Disable Tx power on the uplink standby card	633
View additional card and system information	634
MXK facility protection on uplink cards (2.1.3)	634
Configure line-red uplink ports for concurrent EAPS (2.2.x)	635
Facility protection on the MXK	637
Redundant uplink configuration	637
Equipment protection	637
Single uplink card facility protection	637
Facility protection	637
Configure card redundancy with the line-red command	638
Prepare an uplink port for EAPS configuration	638
EAPS	640
Recommendations for success using EAPS	642
Creating asymmetric and TLS EAPS rings	643
Asymmetric EAPS	643
TLS EAPS	646
Common EAPs topologies	649
EAPS topology command	650
eaps topo	650
eaps topo2	653
Configure line-red state for concurrent EAPS ports (2.2.x and later)	656
Managing inband using IP on a bridge with EAPS	657
Management on an asymmetric EAPS ring	657
Management on a TLS EAPS ring	658
IP applications using IP on a bridge with EAPS	660
EAPS commands	664
Displaying and updating Ethernet interfaces	668
Small form factor pluggables	670
Uplink card pinouts	670
Serial (craft) port pinouts	670
Ethernet port pinouts	671
 Chapter 10 MXK GPON Cards	 673
GPON cards	675
GPON card overview	675
GPON card specifications	676

GPON card configuration	677
View additional card and system information	678
GPON on the MXK	679
GPON terminology	679
Components of GPON optical deployment networks	679
Relationship between T-ports and GEM ports.....	680
Bridge add commands with ranges of Slots, OLTs, GEM ports, and UNI ports..	682
Planning GPON networks	689
Installation testing	690
Handling fiber	691
Smart OMCI GPON zNID installation	692
OMCI overview	693
Smart OMCI overview.....	693
OMCI Profiles	693
Dynamic GEM ports	695
OMCI GPON zNID installation with Smart OMCI	696
Create a ME profile through SMART OMCI web-interface	697
Download a ME profile file to the MXK	701
Create a ME profile for the selected ONT model	702
Create Generic profiles for service plan.....	702
Create high speed Internet on GPON OMCI on uplink and downlink bridges	706
Create uplink and downlink bridges on GPON OMCI for video.....	710
Create uplink and downlink bridge on GPON OMCI for VoIP.....	713
Delete the OMCI profile	717
Import and export the OMCI profile.....	720
Unified Service Provisioning GPON zNID installation.....	725
CPE Menu System	725
One GEM port Allocated for Internal Communication with the ONT for USP ...	728
GEM Ports Assignments in USP	729
Auto Assigned GEM ports	729
Arbitrary GEM ports	729
GPON Traffic Profile Assignment in USP	731
Auto Assigned GTPs	731
Manual Specified GTPs.....	734
Dynamic OMCI GPON zNID Installation.....	734
Dynamic OMCI Overview	735
OMCI GPON zNID Installation with Dynamic OMCI for Triple Services...	748
Viewing all services on an ONU	789
Deletion of CPE profiles and CPE connection that associated on an ONU...	790
Residential Gateway (RG) Features Provisioning	791
RG Provisioning Overview	792
OMCI GPON zNID with RG Features Installation for Triple Services.....	799
CPE System Level Default Settings.....	827
CPE WAN Level IP-Common Settings	830
CPE LAN Level IP-Common Settings.....	832
Static configuration on the WAN side interfaces (without DHCP)	834
Static configuration on the LAN side interfaces with a new DHCP server ...	836
Configuration of Static Routes	839
Configuration of DNS Hosts and DNS Proxy	841

Configuration of Firewall	844
Configuration of DHCP Server	849
Configuration of Conditional DHCP server	850
Configuration of PPPoE username and password	853
Configuration of TR-069.....	855
Set factory default for an ONU	856
System Name and Location of zNID	857
Guided VLAN	858
PoE Power Control per Port & Total Power Budget	858
Power Shedding Enable/Disable Per Port	859
AutoConfiguration and AutoDiscovery OMCI GPON zNID Installation.....	861
Overview	863
OMCI GPON zNID installation with AutoConfiguration and AutoDiscovery for tripleplay services	864
VoIP Phone with LLDP-MED Network Policy.....	880
LLDP	880
LLDP-MED.....	881
Additional Features in Unified Service Provisioning with “bridge add” Command.....	885
VLAN translation on ONU	885
DSCP to COS mapping	889
Support UNI range in “bridge” command.....	891
Support RG CoS in “bridge” command	896
Create an RG-bridged connection without LAN members	897
Create an RG connection without creating a VLAN in RG	898
Post Configuration in USP.....	898
ONU Software Upgrades.....	900
ONU Software Upgrades via OMCI.....	900
Manual upgrade on an ONU	900
Auto upgrade on an ONU.....	904
View the ONU upgrade status.....	907
ONU Software Upgrades via TFTP/SNMP.....	909
Manage ONU with OMCI.....	910
Monitoring ONU Status and Alarms	910
Rebooting, Resyncing and Reprovisioning of ONUs	912
Reboot an ONU	913
Re-synchronize an ONU	913
Re-apply an ONU	913
Monitoring ONU UNI ports Status and Alarms, Configuring ONU UNI port Admin Status and Port speed.....	913
Retrieve status of subscriber facing ports.....	914
Retrieve alarm information on an ONU	914
Administration of subscriber facing ports	914
Configurable speed of subscriber facing ports.....	915
Updating the System Time on the MXK and ONUs	916
Deleting ONU configuration.....	916
Moving ONU configuration.....	919
Cloning ONU configuration	920
MXK GPON using the Reg ID for provisioning	923
Configuring Reg ID	923

Bandwidth Allocation for Upstream Traffic from the ONU to the MXK	925
Configure GPON traffic profile	925
Dynamic Bandwidth Allocation (DBA)	934
GEM port creation	938
Create a GEM port	938
View the GEM port-related information	941
Locate the ONU with its GEM port	942
GEM port level encryption	943
GPON ONU serial number format (Hexadecimal or Decimal)	945
Associate a vendor ID and a serial number with an ONU when activating the ONU	946
Received Signal Strength Indication (RSSI) and Digital Diagnostic Monitoring (DDM)	948
Configurable range for Reserved VLAN per GEM port	951
Configuring the VLAN block	952
Planning for GEM ports	954
GPON type B redundancy	956
Switchover between active and standby OLT ports	962
Automatically switched from active to standby	962
Manually switched from active to standby	963
Manually switched from standby to active	963
GPON redundancy configuration limitations	963
GPON extended reach	965
Recommendations for extended reach	965
Command to measure the distance between MXK and ONT	966
Commands to enable extended reach	966
GPON Business Applications	968
Multicast VPN point-to-point service support on a wire bridge for GPON	968
Upstream multicast video support	968
ONT Inventory Report	970
OMCI Statistics	972
PON Statistics	975
View OLT statistics	975
View ONU statistics	983
GPON Alarms and Traps	987
GPON Alarms	987
Monitor GPON alarms	987
GPON BIP Threshold Crossing Monitor Alarms	987
GPON High and Low Receive Power Threshold Alarms	992
Rogue ONU detection and rogue ONU alarms	995
ONU Dying Gasp Alarms	1007
ONU Manual Reboot Alarms	1008
GPON Traps	1010
View or change trap reporting status on an ONU	1010
Change alarm severity for LineStatusTraps	1011

Chapter 11 MXK VDSL2 Cards	1013
VDSL2 24-port single slot cards	1013
VDSL2 24-port card overview.....	1014
VDSL2 card specifications	1015
VDSL2 24-port card configuration.....	1016
View additional card information	1019
VDSL2 48-port single slot card	1020
VDSL2 48-port line card overview.....	1021
VDSL2 48-port with vectoring	1021
VDSL2 48-port card specifications	1022
VDSL2 48-port card configuration.....	1023
Cabling for the VDSL2 48-port card	1025
VDSL2 on the MXK	1027
VDSL2 overview	1027
VDSL2 standards.....	1027
VDSL2 transmission.....	1027
VDSL2 on the MXK	1028
VDSL2 interfaces	1029
VDSL2 interface profiles.....	1029
vdsl-config default parameters.....	1029
vdsl-co-config default parameters.....	1034
View vdsl-cpe-config profile default parameters	1041
Configure VDSL2 profiles to cap train rates	1048
Configure VDSL2 G.INP.....	1049
VDSL2 statistics	1051
View VDSL2 statistics	1051
View VDSL2 statistics for vectoring	1053
View VDSL2 statistics with the -v variable	1054
Clear VDSL2 counters	1056
VDSL statistics parameters	1057
ADSL2+ fallback for VDSL2	1065
ADSL2+ fallback for VDSL2 overview	1065
Case 1: single-service on untagged downlink bridge configurations	1066
Case 2: single-service on tagged downlink bridge configurations	1067
Case 3: non-default vpi/vci single-service bridge on tagged or untagged downlink..	1068
Case 4: multi-services on tagged downlink bridge configurations.....	1072
Case 5: multi-services on tagged and untagged bridges with non-default vpi/vci	1074
Case 6: multi-services on tagged bridges for ADSL PTM and VDSL PTM.....	1077
ADSL2+ and VDSL2 bonding.....	1079
ADSL2+ and VDSL2 bonding rules on 24-port and 48-port VDSL2 cards	1079
24-port VDSL2 DSP core boundaries and bonding rules	1079
48-port VDSL2 DSP core boundaries and bonding rules	1080
Bonding rules common to the 24-port and the 48-port VDSL2 card.....	1081
Create gbond groups for VDSL2	1082
Bond group creation on 24-port VDSL2 card	1083
Bond group creation on 48-port VDSL2 card	1084

Bridging on ADSL2+ bonding for ADSL	1085
Bridging on ADSL2+ bonding for ADSL	1086
Update the vdsl-config file for gbond group members for ADSL2 modems	1086
Create a tagged downlink bridge on gbond groups with vpi/vci and VLAN ID..	1088
Create a TLS bridge with vpi/vci and VLAN ID	1089
Bridging on VDSL2 bonding.....	1089
Update the vdsl-config file for gbond group members for VDSL2 modems	1089
Create a tagged downlink bridge on gbond groups with VLAN ID	1092
Create a tagged TLS bridge on gbond groups with VLAN ID	1093
Bridging on ADSL2+ bonding for ADSL	1095
Update the vdsl-config file for gbond group members for ADSL2 modems	1095
Create a tagged downlink bridge on gbond groups with vpi/vci and VLAN ID	1096
Create a TLS bridge on gbond groups with vpi/vci and VLAN ID.....	1097
Bridging on VDSL2 bonding for VDSL	1099
Update the vdsl-config file for gbond group members for VDSL2 modems	1099
Create a tagged downlink bridge on gbond groups with VLAN ID	1102
Create a tagged TLS bridge on gbond groups with VLAN ID	1102
Upstream Power Backoff (UPBO) for VDSL2	1104
Downstream Power Backoff (DPBO)	1106
Example calculating E-Side Cable Model parameters.....	1110
VDSL2 statistics	1116
View VDSL2 statistics.....	1116
View VDSL2 stats for vectoring.....	1117
View VDSL2 statistics with the -v variable	1117
Clear VDSL2 counters	1119
VDSL statistics parameters.....	1119
VDSL2 24-port card pinouts	1126
VDSL2 48-port card pinouts	1127
 Chapter 12 MXK Active Ethernet Cards	1131
20-port Active Ethernet dual-slot card	1131
Active Ethernet dual-slot card overview.....	1132
Active Ethernet dual-slot card specifications	1133
Active Ethernet dual-slot card configuration.....	1133
View additional card and system information	1135
20-port Active Ethernet single-slot card	1137
Active Ethernet single-slot card overview.....	1137
Active Ethernet single-slot card specifications.....	1138
Active Ethernet single-slot card configuration	1138
View additional card and system information	1140
20-port Active Ethernet single-slot card with C-SFP support	1142
Active Ethernet single-slot card with compact SFP support overview.....	1142
Active Ethernet single-slot card with compact SFP support specifications	1143
Active Ethernet single-slot card with compact SFP support configuration.....	1143

View additional card and system information	1145
10-port Active Ethernet single-slot card with 2X10G-8XGE	1147
MXK-AE-2X10G-8X1GE line card overview	1147
MXK-AE-2X10G-8X1GE specifications	1148
MXK-AE-2X10G-8X1GE configuration	1148
Link aggregation on the MXK-AE-2X10G-8X1GE line card	1151
SFPs and SFP+s on the MXK-AE-2X10G-8X1GE line card	1151
Displaying and updating Ethernet interfaces	1152
Small form factor pluggables	1154
Ethernet redundancy	1154
Create Ethernet line redundancy	1155
Create a downlink bridge interface on redundant Ethernet ports	1157
Create bridge interfaces on redundant Ethernet ports for intralink configurations	1158
Create bridge interfaces on redundant Ethernet ports for TLS configurations	1159
Removing redundant Ethernet ports	1161
Switchover from active to standby Ethernet port	1162
Automatically switched	1162
Manually switched	1162
Ethernet redundancy configuration limitations	1162
Port redundancy on Active Ethernet line cards	1164
Default Ethernet alarms on line card Minor	1165
Settable alarm severity for Ethernet ports	1165
Enhanced Ethernet port statistics	1168
 Chapter 13 MXK ADSL2+ Bond Cards	1185
ADSL2+ bond cards	1185
ADSL2+ bond 48-port card overview	1186
ADSL2+ bond 48-port card specifications	1187
ADSL+POTS combo card configuration	1190
Internal line testing	1193
ADSL2+ bond 48-port card configuration	1193
View additional card information	1195
ADSL2+ bond 72-port card overview	1196
ADSL2+ bond 72-port card specifications	1197
ADSL2+ bond 72-port card configuration	1198
View additional card information	1200
ADSL2+ on the MXK	1202
ADSL2+ overview	1202
ADSL2+ transmission modes	1203
ADSL2+ rate adaptation	1203
Advanced ADSL2+ configurations on the MXK	1204
Fine tuning ADSL2+ video performance	1204
Seamless Rate Adaptation	1207
Transport mode: fast or interleaved	1209
ADSL2+ interface configuration	1213

ADSL2+ interface overview	1213
View adsl-profile parameter defaults	1214
View adsl-co-profile parameter defaults	1217
View adsl-cpe-profile parameter defaults	1226
Upstream and downstream tone ranges	1234
Configure ADSL2+ profiles for Annex M in fast mode	1235
Configure ADSL2+ profiles for Annex M in interleaved mode	1238
Configure ADSL2+ profiles for G.lite	1241
Configure ADSL2+ profiles to cap train rates	1244
Configure ADSL2+ S=1/2	1249
Configure Broadcom Phy-R™ parameters	1255
Configure G.INP parameters	1257
ADSL2+ statistics	1259
ADSL2+ 48-port bonding	1272
ADSL2+ 72-port bonding	1276
Create gbond groups on 72-port ADSL cards	1277
Delete bond groups	1278
ADSL2+ POTS line card ATM	1279
ATM data	1279
VPI and VCI ranges	1279
Service categories	1279
Constant Bit Rate (CBR)	1279
Non-real-time variable bit rate (nrt-VBR)	1279
Real-time variable bit rate (rt-VBR)	1280
Unspecified bit rate (UBR)	1280
Traffic descriptors	1280
Traffic descriptor parameters	1280
ATM sample configurations	1281
ATM traffic descriptor example for data	1281
ATM traffic descriptor example for video	1281
ATM statistics	1281
ADSL2+ statistics	1283
ADSL2+ Cabinet Mode	1295
Setting cabinet mode	1295
Downstream Power Backoff (DPBO)	1299
ADSL2+ cable and port pinouts	1299
ADSL2+ bond 48-port card pinouts	1299
ADSL2+ bond 48-port card cable pinouts	1303
ADSL-48 to dual 50-pin connector cable	1303
ADSL 48-port card to dual 50-pin connector cables	1308
Variations of ADSL2+ bond 48-port to dual 50-pin connector cables	1309
ADSL2+ bond 72-port card pinouts	1310
ADSL2+ bond 72-port card cable pinouts	1315
dual 78-pin to dual 78-pin connector cable	1316
dual 78-pin to three 50-pin connector cable	1323
dual 78-pin to blunt connector cable	1331
ADSL2+ testing (SELT/DELT) on the MXK	1334

SELT (Single-End Loop Test)	1334
DELT (Dual-End Loop Test)	1339
Chapter 14 MXK POTS Cards	1345
P-phone POTS 24 card (MXK-POTS-EBS-PKT-24)	1346
POTS 72 card (MXK-POTS-72)	1348
POTS card configuration	1350
Configuring 24-port POTS EBS cards	1350
Configuring a POTS-EBS card for packet voice	1351
Configure a 72-port POTS card	1358
Verifying the slot card installation	1360
ADSL+POTS combo cards (MXK-ADSL2+-POTS-BCM-48A-2S, MXK-ADSL2+-POTS-BCM-48A-RNG-2S)	1361
ADSL+POTS combo card configuration	1363
VDSL2+POTS combo card (MXK-VDSL2-POTS-BCM-17A-24)	1366
VDSL+POTS combo card configuration	1367
POTS interface configuration	1370
Internal line testing and ring usage	1375
POTS 24-port cards pinouts	1376
POTS 72-port cards cable and port pinouts	1378
POTS 72-port card port pinouts	1378
POTS 72-port card cable pinouts	1384
Dual 78-pin to dual 78-pin connector cable	1384
Dual 78-pin to three 50-pin connector cable	1391
Dual 78-pin to blunt connector cable	1399
Chapter 15 MXK EFM SHDSL Cards	1403
EFM SHDSL cards	1403
EFM SHDSL card overview	1404
EFM SHDSL card specifications	1405
EFM SHDSL-24 card configuration	1406
Enter a card-profile for the card	1406
Set wetting current	1408
Switch clocking source	1409
MXK EFM SHDSL bonding overview	1410
G. SHDSL bond group configuration	1411
Conditions and limitations for cross-card bonding	1411
Bond group bandwidth specifications	1411
Bond group configuration	1412
EFM auto bonding	1412
EFM manual bond groups	1414
Create bond groups on one card	1414
View bond groups	1416

Change bond group type	1417
Move bond group members	1418
Delete bond groups	1419
Cross-card bonding	1420
SHDSL error monitoring	1420
SHDSL error monitoring statistics	1421
SHDSL error monitoring fields	1421
Auto-bond type switching	1423
Configure the pme-profile	1423
Configure automatic baud rate adaption and fixed rate settings	1424
Configure auto-negotiate or specific data rate	1425
Configure constellation for a TCPAM setting	1426
Set a region	1428
SNR monitoring for bonded G.SHDSL lines	1430
SNR monitoring for the MXK	1430
SNR monitoring for the MXK overview	1430
Current condition SNR maximum threshold	1431
Current condition minimum SNR threshold	1431
MXK SNR monitoring pme-profile parameters	1431
Usage for SNR pme-profile and efm-port parameters	1433
MXK SNR monitoring configuration	1434
Set SNR for target current condition or target worst case mode	1434
Set MXK time and day	1435
Set SNR monitoring from the CLI	1435
View SNR monitoring statistics	1438
Set SNR monitoring in the pme-profile	1439
Configure SNR crossing traps	1442
Verify SNR monitoring is enabled/disabled	1442
G. SHDSL SNR monitoring example	1443
Disable SNR monitoring	1448
SHDSL error monitoring	1449
SHDSL error monitoring statistics	1449
SHDSL error monitoring fields	1449
SHDSL statistics	1452
Bond group statistics and port statistics	1456
View port statistics	1456
View bond group statistics	1457
EtherXtender statistics	1458
802.3ah EFM OAM	1463
MXK-EFM-SHDSL-24 pinouts	1465
Power and data connections for SHDSL CPE devices	1467
Deliver power and data to the CPE	1467
Enable power on the SHDSL line	1469
MTAC testing	1470

Chapter 16	MXK EFM T1/E1 Card	1471
	EFM T1/E1 card overview	1472
	EFM T1/E1 card specifications	1473
	EFM T1/E1 card configuration	1474
	Create a card-profile for the EFM T1/E1 card	1474
	Activate a Ds1 interface	1477
	View the Ds1 interface	1477
	Net-to-net bonding	1484
	EFM auto bonding	1484
	Display bond groups	1484
	Create bond groups from the CLI	1485
	Delete bond groups	1487
	Bond group statistics and port statistics	1488
	Display statistics for an T1/E1 port	1488
	Display statistics for a bond group	1491
	EFM T1/E1 24-port cables	1493
	MALC-CBL-T1/E1-2-45DEG	1493
	Blunt cables	1497
	Tests on the EFM T1/E1 card	1502
	T1/E1 Test Access	1502
	Bit Error Rate Testing (BERT)	1502
	BERT for T1 EFM	1504
Chapter 17	MXK T1/E1 Pseudo Wire Emulation (PWE) Card	1509
	PWE T1/E1 24-port line card	1509
	PWE T1/E1 24-port line card overview	1510
	PWE T1/E1 24-port line card specifications	1511
	PWE T1/E1 24-port line card configuration	1511
	Testing T1/E1	1512
	T1/E1 24 port TDM cables	1514
	MXK-CBL-T1/E1-2-45DEG	1514
	T1/E1 24 blunt cables	1518
Chapter 18	MXK Test Access Cards	1523
	TAC cards	1523
	TAC card overview	1524
	TAC card specifications	1525
	Connectors on the TAC cards	1526
	Metallic loop testing	1527
	Internal look out line test	1527
	Cards supporting look-out test access	1528
	Ring generator	1528
	Configure TAC cards	1530

Creating card profiles for TAC cards.....	1530
Performing line test using TAC cards with external testing set.....	1533
Connecting the external test set to TAC card	1533
Connecting the test measurement device to the metallic test access port.....	1534
Connecting a console to the external test set control port	1535
Performing internal line test with TAC-ITM-RING card	1537
Working with the TAC line test command	1537
Test IDs	1539
Metallic loop tests	1541
3 elements capacitance test.....	1542
3 elements insulation resistance test.....	1543
DC feed self-test.....	1544
DC loop resistance test	1545
Distance to open test.....	1546
DTMF and pulse digit measurement test	1546
Foreign AC currents test.....	1548
Foreign DC voltage test.....	1548
Foreign AC voltage test.....	1549
Howler test	1550
Metering self test	1550
Noise test	1551
On-Off hook transition test.....	1551
Loop and battery condition test	1552
Receiver off-hook test	1553
Ringer equivalency number test.....	1554
Ringing self test.....	1554
Ringing monitor test.....	1555
Tone generation test	1555
Trans-hybrid loss test	1556
Transmission self test	1556
Troubleshooting with metallic loop tests	1557
Auto-calibration	1560
Lookout block diagram	1560
Configuring external alarms	1562
Configuring an external clock.....	1562
Connecting an external ring source	1565
TAC cards pinouts.....	1568
External ring generator input port pinouts	1568
External alarm sense pinouts	1569
Examples of alarms with specific pinouts	1570
Metallic test access port pinouts	1574
External test set control port pinouts	1576
External clock input port pinouts.....	1576
 Chapter 19 Small Form Factor Pluggable (SFP) Connectors.....	 1579
Small form factor pluggables (SFPs)	1579

SFPs for 10 Gig ports on MXK uplink and Active Ethernet line cards.....	1579
SFPs for 1 GE ports	1580
SFPs for MXK uplink cards	1580
XFPs for MXK uplink cards	1581
SFPs for MXK Active Ethernet line cards.....	1581
Single-channel SFPs.....	1581
Dual-channel SFPs	1581
GPON SFP specifications	1582
Insert and remove a fiber connection and an SFP	1583
Insert and remove a dual bi-directional SFP and fiber connector	1584
View SFP information on the MXK.....	1586
Index.....	1591

ABOUT THIS GUIDE

This guide is intended for use by installation technicians and system and network administrators. It explains how to configure the MXK, provision uplink and line cards, create IP interfaces, configure bridges, and other system administration and networking tasks.

This chapter describes:

- [Style and notation conventions, page 27](#)
- [Typographical conventions, page 28](#)
- [Related documentation, page 28](#)
- [Acronyms, page 29](#)
- [Contacting Global Service and Support, page 30](#)

Style and notation conventions

The following conventions are used in this document to alert users to information that is instructional, warns of potential damage to system equipment or data, and warns of potential injury or death. Carefully read and follow the instructions included in this document.



Caution: A caution alerts users to conditions or actions that could damage equipment or data.



Note: A note provides important supplemental or amplified information.



Tip: A tip provides additional information that enables users to more readily complete their tasks.



WARNING! A warning alerts users to conditions or actions that could lead to injury or death.



WARNING! A warning with this icon alerts users to conditions or actions that could lead to injury caused by a laser.

Typographical conventions

[Table 1](#) describes the typographical styles that this guide uses to represent specific types of information.

Table 1: Typographical styles

Bold	Used for names of buttons, dialog boxes, icons, menus and profiles when placed in body text, and property pages (or sheets). Also used for commands, options, parameters in body text, and user input in body text.
Fixed	Used in code examples for computer output, file names, path names, and the contents of online files or directories.
Fixed Bold	Used in configuration examples for text entered by users.
<i>Italic</i>	Used for book titles, chapter titles, file path names, notes in body text requiring special attention, section titles, emphasized terms, and variables.
PLAIN UPPER CASE	Used for environment variables.
Command Syntax	Brackets [] indicate optional syntax. Vertical bar indicates the OR symbol.

Related documentation

Refer to the following documents for additional information:

MXK Hardware Installation Guide — explains how to configure bridging, GPON, link aggregation, and other configuration tasks.

Zhone CLI Reference Guide — explains how to use the Zhone command line interface (CLI) and describes the system commands and parameters.

Refer to the release notes for software installation information and for changes in features and functionality of the product (if any).

Acronyms

Table 2 provides a description of the acronyms that are related to Zhone products and may be found in this manual.

Table 2: Acronym definitions

Acronym	Description
ARP	Address resolution protocol
ATM	Asynchronous Transfer Mode
IAD	Integrated access device
MALC	Multi-access line concentrator
MIB	Management information bases
OLT	Optical line terminal
ONT	Optical network terminal
ONU	Optical network unit
PBX	Private branch exchange
POTS	Plain old telephone service
RIP	Routing Information Protocol
SFP	Small form factor pluggable
SLMS	Single Line Multi-Service
SNMP	Simple Network Management Protocol
TAC	Test Access Card
TFTP	Trivial File Transfer Protocol
XFP	10 Gigabit Ethernet small form factor pluggable
ZMS	Zhone Management System

Contacting Global Service and Support

If your product is under warranty (typically one year from date of purchase) or you have a valid service contract, you can contact Global Service and Support (GSS) for questions about this or other Zhone products, or for Technical Support or Hardware Repairs.

Before contacting GSS, make sure you have the following information:

- Zhone product you are using
- System configuration
- Software version running on the system
- Description of the issue
- Your contact information

If your product is not under warranty or you do not have a valid service contract, please contact GSS or your local sales representative to get a quote on a service plan. You can view the options on our web site at

<http://www.zhone.com/support/services/warranty>.

Technical support

The Technical Assistance Center (TAC) is available with experienced support engineers who can handle questions, assist with service requests, and help troubleshoot systems.

Hours of operation	Monday - Friday, 8 a.m. to 5 p.m, Pacific (excluding U.S. holidays)
Telephone (North America)	877-ZHONE20 (877-946-6320)
Telephone (International)	510-777-7133
E-mail	support@zhone.com
The Web is also available 24 x 7 to submit and track Service Requests (SR's)	www.zhone.com/support

If you purchased the product from an authorized dealer, distributor, Value Added Reseller (VAR), or third party, contact that supplier for technical assistance and warranty support.

Hardware repair

If the product malfunctions, all repairs must be authorized by Zhone with a Return Merchandise Authorization (RMA) and performed by the manufacturer or a Zhone-authorized agent. It is the responsibility of users requiring service to report the need for repair to GSS as follows:

- Complete the RMA Request form (<http://www.zhone.com/account/sr/submit.cgi>) or contact Zhone Support via phone or email:
Hours of operation: Monday Friday, 6:30am-5:00pm (Pacific Time)
E-mail: support@zhone.com (preferred)
Phone: 877-946-6320 or 510-777-7133, prompt #3, #2
- Provide the part numbers and serial numbers of the products to be repaired.
- All product lines ship with a minimum one year standard warranty (may vary by contract).
- Zhone will verify the warranty and provide the customer with a repair quote for anything that is not under warranty. Zhone requires a purchase order or credit card for out of warranty fees.

1

MXK

This chapter provides an overview of MXK networking and features:

- [MXK overview, page 33](#)
- [MXK chassis cards, page 33](#)
- [MXK specifications, page 39](#)

MXK overview

The MXK platform is an intelligent terabit access concentrator that provides scalable multi-service architecture on the SLMS access operating system.

The MXK, in conjunction with zNIDs, provides a complete end-to-end access solution for fiber deployments (GPON and Active Ethernet) that provide triple-play services to subscribers. zNIDs at customer sites extend network intelligence all the way to subscribers with the ability to fine-tune performance.

MXK uplinks are the primary communication channel between subscribers and upstream networking devices. The MXK uplink cards support both copper and fiber SFPs, link aggregation, link redundancy, and the EAPS ring interface.

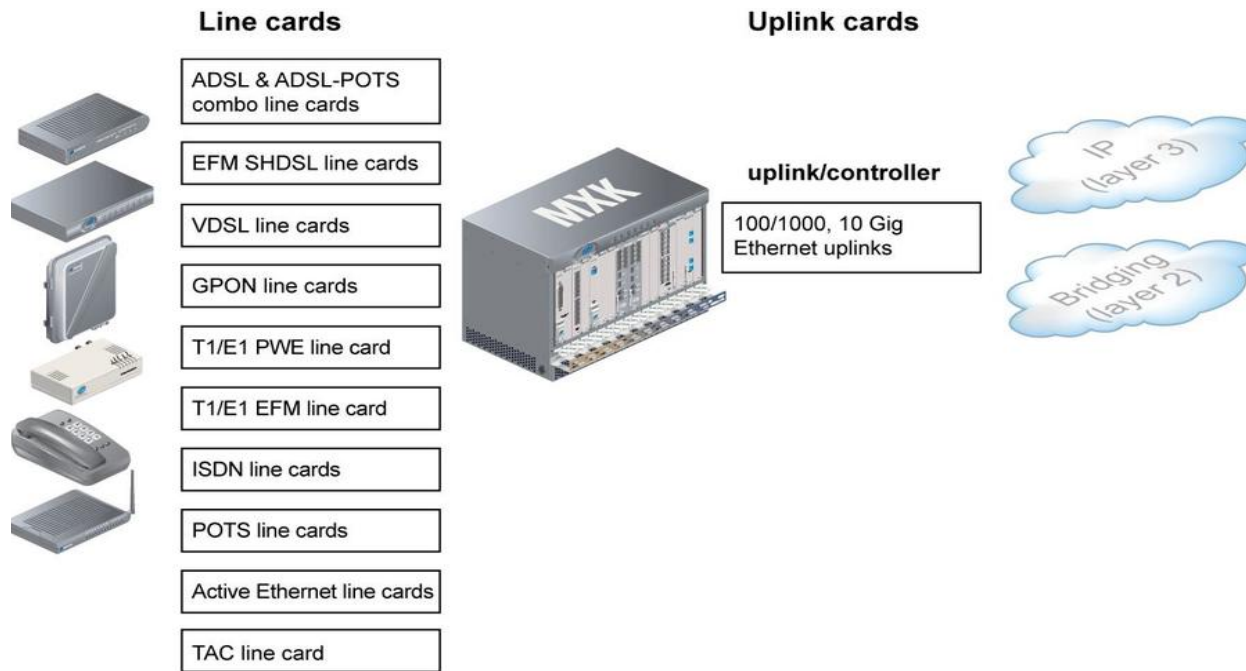
The MXK can be deployed in Central Office environments or outdoor controlled environmental vaults for remote terminal applications. The MXK is intended for restricted access locations only.

MXK chassis cards

The redundant Ethernet uplinks on the MXK enable network providers to provision all classes of services in a single platform and leverage the existing copper infrastructure going to the Digital Loop Carrier (DLC) locations. The variety of MXK line cards offer a wide range of FTTx solutions.

[Figure 1](#) shows the different types of network technologies the MXK supports.

Figure 1: MXK configuration overview



The two types of cards supported on the MXK are uplink cards and line cards.

The MXK has a non-blocking architecture with a high-speed backplane. Each line card on the MXK had a dedicated backplane trace to each of the uplink cards.

The MXK chassis, uplink cards, line cards, and SFPs are temperature hardened.

MXK uplink cards

The MXK uplink cards provide a mix of multiple 10G and 1G interfaces that comply with a variety of network designs. MXK uplink cards provide high-speed Gigabit Ethernet interfaces with active/standby redundancy.

For information on uplink card configuration, see [Chapter 9, MXK Ethernet Uplink Cards, on page 607](#).

The MXK uplink cards are:

- MXK MXK-UPLINK-2X10GE-8X1GE
Two 10 GE and eight 100/1000 Ethernet interfaces, supports all line cards.
- MXK MXK-UPLINK-8X1G
Eight 100/1000 Ethernet interfaces, supports all line cards.
- MXK-UPLINK-4X1GE
Four 100/1000 Ethernet interfaces, supports all line cards.
- MXK-UPLINK-4X1GE-CU

Four 100/1000 Ethernet interfaces, supports only copper line cards.

- MXK-UPLINK-6X1GE-CLK

Six 100/1000 Ethernet interfaces to support all line cards. The CLOCK input port supports TI/E1 or BITS

- MXK-UPLINK-2X10G-8X1G-CLK

Provides high-speed Gigabit Ethernet interfaces with active/standby redundancy and consists of two 10 GE and eight 100/1000 Ethernet interfaces to support all line cards. The CLOCK input port supports TI/E1 or BITS

MXK line cards

The MXK line cards support GPON, Active Ethernet, ADSL2+, G. SHDSL EFM, POTS for VoIP, VDSL2, EFM T1/E1, PWE T1/E1, and TAC.

The MXK line cards are:

- Active Ethernet

MXK-AEX20-FE/GE-2S

A two slot card that supports Ethernet traffic over 20 ports that provide either 100/1000 Base-T, fiber 100FX or 1 Gigabit Ethernet interfaces to support distances as high as 80km depending on the SFPs used.

MXK-AEX20-FE/GE

A slot card that supports Ethernet traffic over 10 ports that provide either 100/1000 Base-T, fiber 100FX or 1 Gigabit Ethernet interfaces to support distances as high as 80km depending on the SFPs used.

MXK-AEX20-FE/GE-CSFP

A slot card that supports multiple subscribers on a single SFP cage through the use of SFPs of type CSFP option 2 with two bi-directional transceivers. This Active Ethernet card also supports single channel SFPs and dual bi-directional (bi-di) SFPs

For information on Ethernet card configuration, see [Chapter 12, MXK Active Ethernet Cards, on page 1131](#).

- GPON

MXK-GPONX4-IO

MXK-GPONX8-IO

A quad or octal interface that supports 2.5 Gbps downstream bandwidth and 1.25 Gbps upstream bandwidth per interface as specified in the G.984.1-4 specifications.

For information on GPON card configuration, see [Chapter 10, MXK GPON Cards, on page 673](#).

- MXK-ADSL2+-BCM-48A

Single slot 48-port card that supports ADSL2+ Annex A/M.

MXK-ADSL2+-POTS-BCM-48A-2S

Two-slot 48-port card that provides integrated ADSL and POTS VoIP service.

MXK-ADSL2+-SPLTR600-BCM-48A-2S

MXK-ADSL2+-SPLTR900-BCM-48A-2S

Two-slot 48-port cards with an integrated POTS splitter to provide ADSL and POTS service. Each of these lines are combined with the ADSL2+ signal internally and exits the line card in the subscriber direction with both ADSL and POTS on the loop. In the network direction the POTS is split from the ADSL signal keeping POTS on copper pairs and placing the ADSL data information on the IP network.

MXK-ADSL2+-BCM-72A

MXK-ADSL2+-BCM-72B

These cards are a single slot card that supports ADSL2+ Annex A/M or ADSL2+ Annex B.

All ADSL cards support VoIP POTS services and support ANSI T1.413 Issue 2, G.992.1 (G.dmt), G.992.2 (G.lite), and ADSL2+ (G.992.5) standards.

For information on ADSL2+ card configuration, see [Chapter 13, MXK ADSL2+ Bond Cards](#), on page 1185.

- MXK-EFM-SHDSL-24-NTP

Single slot 24-port card provides network timing reference and line power.

MXK-EFM-SHDSL-24-NTWC

Single slot 24-port card provides network timing reference and current.

For information on EFM-SHDSL card configuration, see [Chapter 15, MXK EFM SHDSL Cards](#), on page 1403.

- MXK-EFM-T1/E1-24

Single slot 24-port card provides 24 T1/E1 bondable ports.

For information on EFM-T1/E1 card configuration, see [Chapter 16, MXK EFM T1/E1 Card](#), on page 1471.

- VDSL

MXK-VDSL2-24-BCM

Single-slot 24-port VDSL2 subscriber line card, which provides high symmetric and asymmetric bandwidth and supports 17a profile.

The MXK-VDSL2-24-BCM card can be used with the Zhone VDSL2 CPE devices. This architecture allows VDSL2 users to access the maximum bandwidth available over twisted-pair, copper phone lines.

MXK-VDSL2-POTS-BCM-17A-24

This card provides 24 ports of integrated VDSL2 and POTS VoIP services and supports SIP, SIP-PLAR, H.248, MGCP protocols, and H.248 (MEGACO) protocols.

MXK-VDSL2--SPLTR600-BCM-17A-24

MXK-VDSL2--SPLTR900-BCM-17A-24

These cards provide integrated POTS splitter to provide 24 ports of integrated VDSL2 and POTS service.

MXK-VDSL2-BCM-17A-48

The MXK-VDSL2-BCM-17A-48 card is single-slot 48-port VDSL2 subscriber line card which provides high symmetric and asymmetric bandwidth and supports up to 17a profile.

MXK-VDSL2-BCM-17A-48-V

The MXK-VDSL2-BCM-17A-48-V card is single-slot 48-port VDSL2 subscriber line card which provides high symmetric and asymmetric bandwidth and supports up to 17a profile.

This VDSL2 card vectoring is a noise-canceling technology that cuts the noise on VDSL2 lines in a bundle allowing the line to operate at peak speeds.

For information on VDSL2 card configuration, see [Chapter 11, MXK VDSL2 Cards, on page 1013](#).

- MXK-PWE-T1/E1-24

Single-slot 24-port PseudoWire Emulation (PWE) card is a circuit emulation service (CES) which supports PWE3 Edge-To Edge Emulation (RFC 3985) over a packet switched network (PSN) and allows T1/E1 circuits to be carried over a PSN.

For information on PWE-T1/E1 card configuration, see [Chapter 17, MXK T1/E1 Pseudo Wire Emulation \(PWE\) Card, on page 1509](#).

- MXK-VDSL2-POTS-BCM-17A-24

Single-slot card that provides 24 ports of integrated VDSL2 and POTS VoIP services.

For information on POTS card configuration, see [Chapter 14, MXK POTS Cards, on page 1345](#).

- MXK-ADSL2+-POTS-BCM-48A-2S

MXK-ADSL2+-POTS-BCM-48A-RNG-2S

Two-slot cards that provide 48-ports of integrated ADSL and POTS VoIP services. These cards support the ANSI T1.413 Issue 2, G.992.1(G.dmt) and G.992.2 (G.lite), G.992.3 and G.992.4 (ADSL2), G.992.5 (ADSL2+), Annex A, and Annex M ADSL standards. Also supported are SIP, SIP-PLAR, MGCP, and H.248 (MEGACO) protocols.

MXK-ADSL2+-POTS-BCM-48A-RNG-2S provides integrated ringing functionality and internal line testing functionality.

For information on POTS card configuration, see [Chapter 14, MXK POTS Cards, on page 1345](#).

- MXK-POTS-EBS-PKT-24

Single slot card that supports POTS or EBS services. This card supports packetized voice service for the POTS and EBS end-users when the MXK chassis is subtended to a MALC with the voice gateway card.

For information on POTS card configuration, see [Chapter 14, MXK POTS Cards, on page 1345](#).

- MXK-POTS-72

A single slot card that supports packetized voice for use in a VoIP network. This card supports loop start, ground start, dial pulse, and provides echo cancellation. It has an integrated ring generator as well as the internal line testing functionality (same capabilities as the enhanced MTAC or TAC ITM card) on the card.

For information on POTS card configuration, see [Chapter 14, MXK POTS Cards, on page 1345](#).

- MXK-MTAC/RING

MXK-MTAC/RING-ENH

A single slot card that supports metallic loop testing for DSL and POTS interfaces with the external test set.

For more information, see [Chapter 18, MXK Test Access Cards, on page 1523](#).

MXK specifications

This section describes some key features of the MXK, including:

Management

The MXK can be managed either in-band (VLAN tagged) on uplink Ethernet ports, out-of-band on the 10/100 Ethernet interface, or IP on a bridge.

The uplink card also contains a serial (craft) port for local management.

After establishing a connection to the MXK, administrators can manage the device using the Command Line Interface (CLI), Web UI, ZMS, or SNMP.

IP and data support

The MXK provides access and aggregation routing functions to connect subscribers to networks. The following MXK interfaces support IP traffic:

- One Ethernet interface on the uplink card only for management.
- High speed Ethernet interfaces on the uplink cards including two 10 GE links and eight 100/1000 Ethernet links.

The MXK provides the following key data services:

- IP forwarding and routing—incoming packets from an interface are forwarded to the appropriate output interface using the routing table rules.
- Bridging—incoming packets from an interfaces are forwarded based on MAC addresses or Layer 2 forwarding rules.
- IP filtering. IP filtering is typically performed to enhance network security by limiting access between two networks.
- Bridging: uplink, downlink, TLS, and intralinks.
- IPv6 is supported for bridging (pass through and bridging related, such as in bridge-paths, video and voice downlinks, PPPoE downlinks)
- IPv4 is supported for bridging (along with IPv6) and for IP services which are terminated on the MXK (management, PWE, POTs ports to softswitch connections).
- Bridging enhancements:
 - IP on a TLS bridge
 - Intralink support including multiple intralinks
 - VLAN wildcard for Q-in-Q
 - DHCP relay
- Routing (uplinks, Active Ethernet)
- Video: Multicast (IGMPv1/v2), IGMP snooping, IGMP proxy reporting

- QoS: rate limiting (three color policing; color blind, 802.1p)
- RIP v1 (RFC 1058) RIPv2 (RFC 2453)
- DHCP server (RFC 2131, 2132)
- Broadcast storm protection
- QoS: Rate limiting, 3 color policing, 802.1p
- Link aggregation
- Q-in-Q (Active Ethernet, GPON)
- Security
 - System security: SSH, HTTPS, and SFTP
 - Secure bridging: Destination MAC swapping, secure bridging filters
- RSTP on uplinks
- GPON
 - Smart OMCI: interoperability with third party ONTs
 - 64 splits, class B+ optics
 - Dynamic GEM port creation

The MXK can be managed with:

- Command line interface (CLI)
- ZMS
- WebUI

Rate Limiting

Rate limiting is a mechanism for controlling traffic and can include policing (dropping packets). Use rate limiting to control the rate of traffic sent or received on the ingress or the egress of both the logical port or the physical port of the MXK. Traffic that is less than or equal to the specified rate is sent and traffic that exceeds the rate is dropped. The rate limiting does not include queuing which delays packets in a buffer.

After configuring an interface with rate limiting, the traffic rate is monitored and metered to verify conformity with an established contract. Non-conforming traffic is discarded, while conforming traffic passes through the interface without any changes. The MXK follows RFC 2697 for rate limiting on both the ingress and egress of the interface.

VoIP

Voice over IP, also known as Internet Telephony, supports full duplex transmission of voice traffic over IP networks. The MXK supports Media gateway control protocol (MGCP) and Session Initiation Protocol (SIP).

MGCP

Media gateway control protocol (MGCP) provides the means to interconnect a large number of IP telephony gateways. MGCP assumes that a call agent (CA) performs the intelligence of all call-control operations and that a media gateway (MG) carries out all media processing and conversion.

The MXK also supports Megaco, H.248.

SIP

Session Initiation Protocol (SIP) is a signaling protocol that provides a mechanism for:

- call establishment
- call teardown
- call control
- other supplementary services in an IP network.

2

MXK OPERATIONS, ADMINISTRATION, AND MAINTENANCE

This chapter describes MXK operations, system administration, and maintenance functions:

- [MXK device management, page 43](#)
- [MXK system administration, page 67](#)
- [MXK port management, page 111](#)
- [MXK security, page 128](#)
- [MXK alarms, page 139](#)
- [MXK card configuration, page 148](#)
- [MXK DNS resolver configuration, page 157](#)
- [CPE Manager, page 158](#)

MXK device management

This section covers MXK device management:

- [Overview of MXK device management, page 43](#)
- [Manage the MXK from the CLI, page 44](#)
- [Manage the MXK from ZMS, page 59](#)
- [Manage the MXK using Zhone Web User Interface, page 64](#)

Overview of MXK device management

In order to access the MXK for management tasks, you must first log into the serial craft port, see [Log into the serial \(craft\) port, page 45](#).

After logging into the MXK, there are three ways to manage the device:

- CLI interface management

See [Manage the MXK from the CLI on page 44](#)

Out-of-band management, see [Out-of-band management on the MXK on page 47](#)

In-band management, see [In-band management on the MXK on page 49](#)

- Zhone Management System (ZMS) remote management

See [Manage the MXK from ZMS on page 59](#)

- Zhone Web UI remote management

See [Manage the MXK from the WebUI on page 63](#)

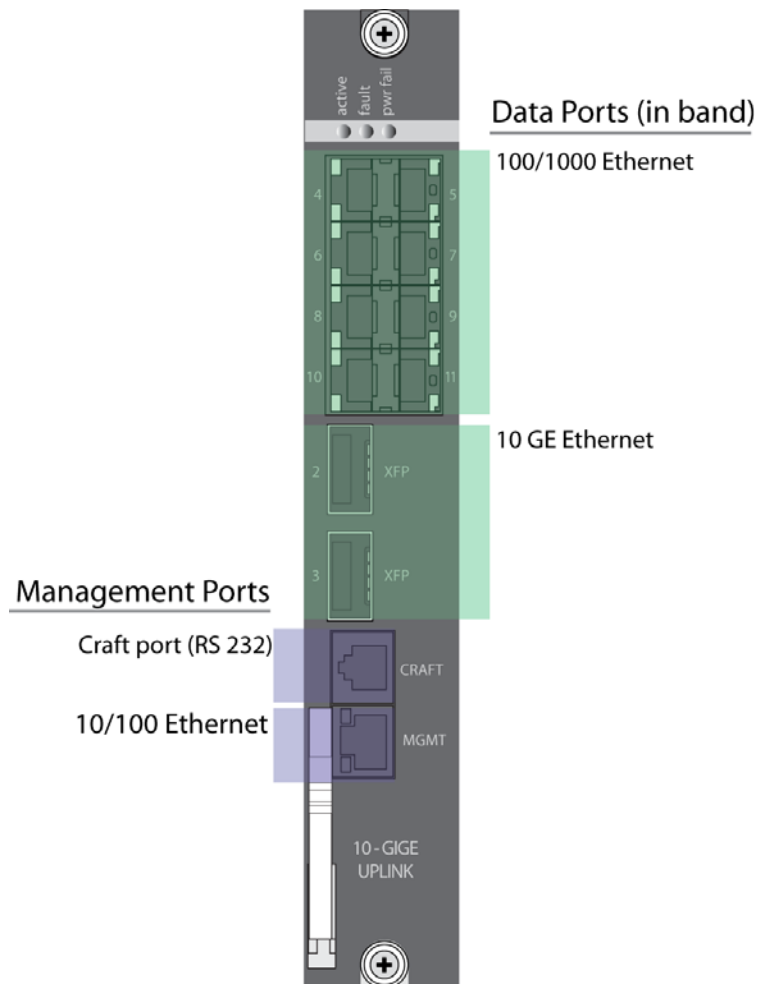
Manage the MXK from the CLI

This section describes how to configure management interfaces on the MXK to access and manage the MXK from the CLI:

There are three ways to manage the MXK, through the serial craft RS 232 port, through the 10/100 Ethernet port (out-of-band management), and through 10 GE or 100/1000 Ethernet ports (in-band management). These ports can be configured for management through the CLI by adding an IP address on either the physical port or on a uplink, TLS, or link aggregation bridge.

[Figure 2](#) shows the ports available for MXK management.

Figure 2: Ports available for MXK management



Log into the serial (craft) port

Log into and out of the system

When you first access the MXK, the default login is **admin** and the default password is **zhone**:

```
login: admin
password:
zSh>
```

To log out of the system, enter the **logout** command:

```
zSh> logout
```



Tip: The system automatically logs you out after a period of inactivity. The default logout time is 10 minutes, but can be changed with the **timeout** command. Refer to the *Zhone CLI Reference Guide* for information on the **timeout** command.

Enabling and disabling logging

By default logging is enabled on the serial craft port and disabled over telnet sessions. To enable or disable logging for the session, using the following command:

```
zSh> log session off | on
```

The **log session** command only applies to the current session. You can also enable or disable logging for all serial craft port sessions using the following command:

```
zSh> log serial on | off
```

This command setting persists across system reboots.

Changing system defaults

The system automatically logs you out after a period of inactivity. The default logout time is 10 minutes.

To change the logout time enter the time-out command with the time in minutes:

```
zSH> timeout 120
CLI time-out value is now at 120 minutes.
```

To turn time-out off enter:

```
zSH> timeout off
CLI timer turned off.
```

To reset time-out to the default enter:

```
zSH> timeout -d
CLI time-out value reset to default of 10 minutes.
```

Using the setline command

The **setline** command sets the maximum lines to be displayed at once.

Entering the **setline** command without an argument displays the current number of lines per page.

```
zSH> setline
lines/page = 19
```

Entering the **setline** command with an argument sets the number of lines displayed per page.

```
zSH> setline 50
cli lines per page changed to: 50
```

View the change.

```
zSH> setline
lines/page = 50
```

Entering the **setline** command with an argument of *0* sets continuous scrolling.

```
zSH> setline 0
0 was entered, setting continuous scroll mode.
```

Out-of-band management on the MXK

This section describes out-of-band management configurations:

- [Configure the serial craft RS 232 port for out-of-band management, page 47](#)
- [Configure an IP interface on the 10/100 BaseT Ethernet port for MXK out-of-band management, page 48](#)



Note: Since the MXK has a passive chassis, you must install the uplink card in slot *a* before you can log in to the serial port and begin the initial configuration of the system.

Configure the serial craft RS 232 port for out-of-band management

The MXK unit provides an out-of-band RS232 D serial (craft) interface for managing the unit. To access the serial port on the uplink card, configure the **rs232-profile** with these settings:

- 9600bps
- 8 data bits
- No parity
- 1 stop bit
- No flow control



Note: Do not use the serial craft port of a standby card to modify its configuration.



Tip: The serial (craft) port settings can be changed by modifying the **rs232-profile**.

You must perform the initial configuration of the system using the serial (craft) interface. After completing the initial configuration, you can manage the MXK unit over the network through a Telnet session over the Ethernet interface.



Note: The MXK supports six concurrent management sessions, five Telnet sessions and a single local session through the serial (craft) port.

Configuring the serial craft RS 232 port for management

Update the **rs232-profile** for the shelf and slot that contain the serial craft port.



Caution: The serial craft port supports speeds of 9600, 19200, 38400, and 57600. Do not set the speed to an unsupported value. Doing so could render the serial craft port inaccessible.

To update the **rs232-profile** enter:

```
zSH> update rs232-profile 1-a-1-0/rs232
rs232-profile 1-a-1-0/rs232
Please provide the following: [q]uit.
rs232PortInSpeed: -----> {9600}: 57600
rs232PortOutSpeed: -----> {9600}: 57600
rs232PortInFlowType: ----> {none}:
rs232PortOutFlowType: ---> {none}:
rs232AsyncPortBits: -----> {8}:
rs232AsyncPortStopBits: -> {one}:
rs232AsyncPortParity: ---> {none}:
rs232AsyncPortAutobaud: -> {disabled}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Configure an IP interface on the 10/100 BaseT Ethernet port for MXK out-of-band management

The MXK has a 10/100 BaseT Ethernet interface on the uplink card for out-of-band management. The **ip-interface-record** profile for this interface is **ethernet1**. This interface is shared between the two Ethernet ports on redundant uplink cards (if they exist). The system can be reached using the address configured in the **ethernet1 ip-interface-record**, no matter which card is active.



Caution: You must configure the Ethernet interface on the uplink card before any other interfaces on the system, even if you do not intend to manage the unit over the Ethernet.

Configuring an out-of-band IP management interface

The following example configures the IP address for out-of-band management of the MXK.

- 1 Configure the 10/100 Ethernet interface on the uplink card.

```
zSH> interface add 1-a-1-0/eth 192.168.8.21/24
Created ip-interface-record ethernet1/ip.
```



Note: Ipv4 is required for all IP termination on the MXK, including management interfaces. IPv6 is not supported for IP termination on the MXK.

2 Verify the interface.

```
zSH> interface show
```

```
1 interface
Interface      Status  Rd/Address      Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 192.168.8.21/24  00:01:47:17:ee:54  ethernet1
-----
```

3 Create the default route.

See [Creating a default route on page 57](#).

In-band management on the MXK

This section describes in-band management on the MXK:

- [Configure IP on a bridge for in-band device management overview, page 49](#)
- [Configure an IP address on a Ethernet uplink port for MXK in-band management, page 50](#)
- [Configure IP on a bridge for Ethernet, page 50](#)
- [Configure TLS IP on a bridge, page 52](#)
- [Configure IP on a bridge on a link aggregation bridge, page 54](#)
- [Configure VoIP on IP on a bridge for EAPS, page 57](#)
- [Create a default route, page 57](#)

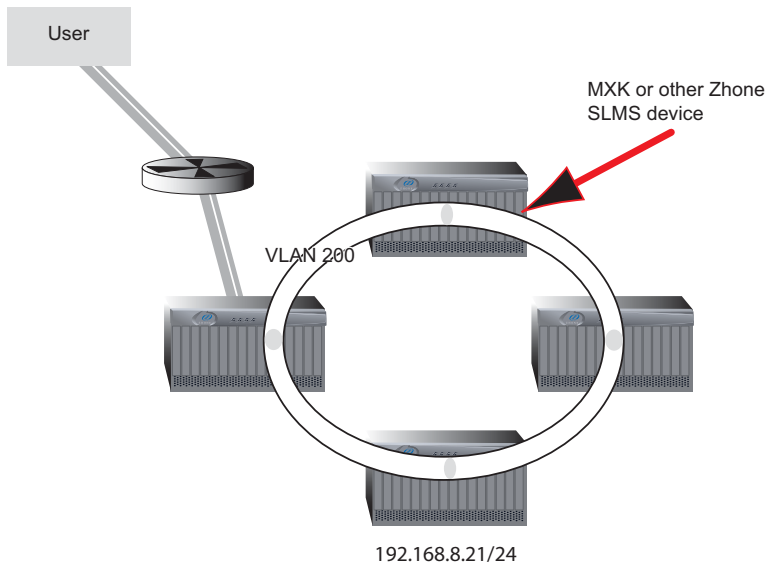
Configure IP on a bridge for in-band device management overview

IP on a bridge allows you to put an IP address on a bridged VLAN for in-band management of the MXK. This VLAN can be used to manage multiple MXKs or other devices. The MXK supports up to six IP on a bridge interfaces per chassis.



Note: Ipv4 is required for all IP termination on the MXK, including **ipobridge** interfaces. IPv6 is not supported for IP termination on the MXK.

Figure 3: IP on a bridge



Configure an IP address on a Ethernet uplink port for MXK in-band management

Configure an IP interface on an uplink port for in-band MXK management.

Configure IP on a bridge for Ethernet

This example creates an IP on a bridge interface using the IP address of 192.168.8.21/24, and a logical port interface 6 on VLAN 200.

Creating IP on a bridge on a uplink bridge for Ethernet

- 1 Create an *uplink* bridge with a VLAN ID.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 200
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-200/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
upl              Tagged 200    1/a/2/0/eth    ethernet2-200/bridge    UP  S VLAN 200 default
1 Bridge Interfaces displayed
```

- 2 Enter **interface add interface/type** with the type as *ipobridge*.

This command creates the new IP interface as well as the new *ipobdwn* bridge. The bridge created will be a subscriber facing downlink bridge.



Note: The logical port interface for IP on a bridge on the MXK must be 1-a-6-0/ipobridge for correct transmission of IP packets.

```
zSH> interface add 1-a-6-0/ipobridge vlan 200 192.168.8.21/24
Created ip-interface-record ipobridge-200/ip.
```

The uplink card is now reachable from the upstream, and IP 192.168.8.21/24 can reach other upstream devices on the same VLAN ID.

Follow the same steps to create an IP on a bridge and bridges for downstream devices.

3 Verify the *ipobridge* interface:

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 10.51.1.118/24    00:01:47:19:b9:78   ethernet1
1/a/6/0/ip     UP      1 192.168.8.21/24   00:01:47:93:74:54   ipobridge-200
-----
```

4 Verify the *ipobridge* and the *uplink* bridge:

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical          Bridge              St  Table Data
-----
upl           Tagged 200    1/a/2/0/eth      ethernet2-200/bridge  UP S VLAN 200 default
ipobdwn       Tagged 200    1/b/6/0/ipobridge ipobridge-200/bridge  UP S 00:01:47:93:74:54
                                     S 192.168.8.21

2 Bridge Interfaces displayed
```

The downlink bridge with the same VLAN ID was automatically created.

5 Create the default route.

See [Creating a default route on page 57](#).

Deleting the IP on a bridge management interface

1 View the IP interface.

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 10.51.1.118/24    00:01:47:19:b9:78   ethernet1
1/a/6/0/ip     UP      1 192.168.8.21/24   00:01:47:93:74:54   ipobridge-200
-----
```

2 Delete the *ipobridge* interface.

```
zSH> interface delete 1/a/6/0/ip
Delete complete
```

This action automatically deletes the *ipobridge downlink* bridge.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge
-----
upl                Tagged 200    1/a/2/0/eth    ethernet2-200/bridge
1 Bridge Interfaces displayed
```

3 Delete the *uplink* bridge.

```
zSH> bridge delete ethernet2-200/bridge vlan 200
Bridge-path deleted successfully
ethernet2-200/bridge delete complete
```

Configure TLS IP on a bridge

This example creates an IP on a bridge interface using the IP address of 192.168.8.21/24 on VLAN 200.

Creating IP on a bridge for a TLS bridge

1 Create a *tls* bridge with VLAN ID.

```
zSH> bridge add 1-a-2-0/eth tls vlan 700
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge
-----
tls                700        1/a/2/0/eth    ethernet2/bridge
1 Bridge Interfaces displayed
```

2 Enter **interface add interface/type** with the type as *ipobridge*.

This command creates the new IP interface as well as a new bridge. The bridge created will be a Transparent Lan Service (TLS) tagged bridge.



Note: The logical port interface for IP on a bridge on the MXK must be *1-a-6-0/ipobridge* for correct transmission of IP packets.

```
zSH> interface add 1-a-6-0/ipobridge vlan 700 192.168.8.21/24
Created ip-interface-record ipobridge-700/ip.
```

The uplink card is now reachable from the upstream, and IP 192.168.8.21/24 can reach other upstream devices on the same VLAN.

Follow the same steps to create an IP on a bridge and bridges for downstream devices.

3 Verify the *ipobridge* interface:

```
zSH> interface show
```

```
2 interfaces
```

Interface	Status	Rd/Address	Media/Dest Address	IfName
1/a/1/0/ip	UP	1 10.51.1.118/24	00:01:47:19:b9:78	ethernet1
1/a/6/0/ip	UP	1 192.168.8.21/24	00:01:47:93:74:54	ipobridge-700

4 Verify the *tls* IP on an bridge interface.

```
zSH> bridge show
```

Type	Orig	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
tls		700		1/a/2/0/eth	ethernet2/bridge	UP	
ipobtls		Tagged	700	1/a/6/0/ipobridge	ipobridge-700/bridge	UP	S 00:01:47:93:74:54 S 192.168.8.21

```
2 Bridge Interfaces displayed
```

The *ipobridge* creates a static IP address and MAC address.

```
zSH> bridge-path show
```

VLAN/SLAN	Bridge	Address
700	N/A	VLAN, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0,
IGMP DSCP: 0, Flap Mode: Fast		
700	ipobridge-700/bridge	192.168.8.21
700	ipobridge-700/bridge	00:01:47:93:74:54

5 Create the default route.

See [Creating a default route on page 57](#).

Deleting the IP on a bridge configuration**1** Verify the IP on a bridge interface.

```
zSH> interface show
```

```
2 interfaces
```

Interface	Status	Rd/Address	Media/Dest Address	IfName
1/a/1/0/ip	UP	1 10.51.1.118/24	00:01:47:19:b9:78	ethernet1
1/a/6/0/ip	UP	1 192.168.8.21/24	00:01:47:93:74:54	ipobridge-700

2 Delete the IP on a bridge interface.

```
zSH> interface delete 1/a/6/0/ip
```

```
Delete complete
```

This action automatically deletes the subscriber facing *ipobridge* *tls* bridge.

```
zSH> bridge show
```

Type	Orig		Physical	Bridge	St	Table Data
	VLAN/SLAN	VLAN/SLAN				
tls		700	1/a/2/0/eth	ethernet2/bridge	UP	
1 Bridge Interfaces displayed						

3 Delete the *tls* network facing bridge.

```
zSH> bridge delete ethernet2/bridge vlan 700
Bridge-path deleted successfully
ethernet2/bridge delete complete
```

Configure IP on a bridge on a link aggregation bridge

This example creates an IP on a bridge interface using the IP address of 192.168.8.21/24, and a logical port interface 6 on VLAN 200.

If you need to create a link aggregation group, see [Chapter 8, Link Aggregation Configuration](#) for link aggregation configuration rules and information.

Creating IP on a bridge on a link aggregation bridge

1 Verify the link aggregation.

```
zSH> linkagg show
LinkAggregations:
slot unit ifName          partner: Sys          Pri    grp ID  status  agg mode
-----
a*  1  1-a-1-0      00:00:00:00:00:00    0x0    0x0    OOS    Active
    links      slot  port  subport          status
    -----
    1-a-2-0      a    2    0              ACT
b   1  1-b-1-0      00:00:00:00:00:00    0x0    0x0    OOS    Active
    links      slot  port  subport          status
    -----
    1-b-2-0      b    2    0              DSA
global linkagg group red type: red
```

2 Create a *linkagg uplink* bridge. The uplink ports are the ports that are in the link aggregation.

```
zSH> bridge add 1-a-1-0/linkagg uplink vlan 200 tagged
Adding bridge on 1-a-1-0/linkagg
Created bridge-interface-record linkagg-a-1-200/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
upl   Tagged 200  1/a/1/0/linkagg  linkagg-a-1-200/bridge  DWN S VLAN 200 default
1 Bridge Interfaces displayed
```

The uplink card is now reachable from the upstream, and IP 192.168.8.21/24 can reach other upstream devices on the same VLAN.

Follow the same steps to create an IP on a bridge and bridges for downstream devices.

3 Enter **interface add interface/type** with the type as *ipobridge*.

This command creates the new IP interface as well as a new bridge. The bridge created will be a *downlink* tagged bridge.



Note: The logical port interface for IP on a bridge on the MXK must be *1-a-6-0/ipobridge* for correct transmission of IP packets.

```
zSH> interface add 1-a-6-0/ipobridge vlan 200 192.168.8.21/24
Created ip-interface-record ipobridge-200/ip.
```

The uplink card is now reachable from the upstream, and IP 192.168.8.21/24 can reach other upstream devices on the same VLAN.

Follow the same steps to create an IP on a bridge and bridges for downstream devices.

4 Verify the interface.

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 10.51.1.118/24    00:01:47:19:b9:78   ethernet1
1/a/6/0/ip     UP      1 192.168.8.21/24   00:01:47:93:74:54   ipobridge-200
-----
```

5 Verify the *ipobridge*.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical          Bridge              St  Table Data
-----
upl   Tagged 200      1/a/1/0/linkagg  linkagg-a-1-200/bridge  DWN S VLAN 200 default
ipobdwn Tagged 200      1/a/6/0/ipobridge ipobridge-200/bridge  UP  S 00:01:47:93:74:54
                                   S 192.168.8.21
```

2 Bridge Interfaces displayed

A static IP and MAC address is created on the bridge path.

```
zSH> bridge-path show
VLAN/SLAN  Bridge              Address
-----
200 linkagg-a-1-200/bridge  Default, Age: 3600, MCAST Age: 250, IGMP Query Interval:
0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym
200 ipobridge-200/bridge    192.168.8.21
200 ipobridge-200/bridge    00:01:47:93:74:54
```

6 Create the default route.

See [Creating a default route on page 57](#).

Deleting the IP on a bridge management interface

1 View the IP interface

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address      Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 10.51.1.118/24  00:01:47:19:b9:78   ethernet1
1/a/6/0/ip     UP      1 192.168.8.21/24  00:01:47:93:74:54   ipobridge-200
-----
```

2 Delete the *ipobridge* interface.

```
zSH> interface delete 1/a/6/0/ip
Delete complete
```

This action automatically delete the *ipobridge downlink* bridge.

3 Delete the *linkagg* bridge.

```
zSH> bridge delete linkagg-a-1-200/bridge vlan 200
Bridge-path deleted successfully
linkagg-a-1-200/bridge delete complete
```

Configure VoIP on IP on a bridge for EAPS

When configuring voice on an EAPS ring, you must use the IP address that you enter for the *ipobridge* interface.

Configuring IP on a bridge for voice on an EAPS ring

- 1 Enter **interface add interface/type** with the type as *ipobridge*.

This command creates the new IP interface as well as a new *ipobridge* bridge. Entering the *tls* bridge type means that the *ipobridge* created will be a *tls* bridge.

```
zSH> interface add 1-a-6-0/ipobridge vlan 400 10.10.10.2/30 tls
Created ip-interface-record ipobridge-400/ip.
```

Verify the interface.

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 10.51.1.118/24    00:01:47:19:b9:78   ethernet1
1/a/6/0/ip     UP      1 10.10.10.2/30     00:01:47:93:74:54   ipobridge-400
-----
```

Verify the *ipobridge* that was created.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
ipobtls      Tagged 400    1/a/6/0/ipobridge  ipobridge-400/bridge  UP  S 00:01:47:93:74:54
                                     S 10.10.10.2

1 Bridge Interfaces displayed
```

- 2 Create the default route for the *ipobridge* IP address.

```
zSH> route add default 10.10.10.1 1
```

Create a default route

Regardless of which management IP interface is created, you must also create a default route for that interface.

Creating a default route

Create the default route using the gateway 192.168.8.1 with a cost of 1 (one).

```
zSH> route add default 192.168.8.1 1
```

Verify the route:

```
zSH> route show
```

Destination Routing Table

Dest	Nexthop	Cost	Owner	Fallback

0.0.0.0/0	192.168.8.1	1	STATICLOW	
192.168.8.0/24	1/a/6/0/ip	1	LOCAL	

Use the **ping** command to verify connectivity to the default gateway:

```
zSH> ping 192.168.8.1
PING 192.168.8.1: 64 data bytes
!!!!
----192.168.8.1 PING Statistics----
5 packets transmitted, 5 packets received
round-trip (ms)  min/avg/max = 0/0/0
```

To stop the ping, press CTRL+C.

Manage the MXK from ZMS

This section describes:

- [Configure the MXK to run ZMS in SNMPv3, page 59](#)
- [Mass provisioning from the CLI when running ZMS, page 60](#)

Before using Zhone Management System (ZMS), the Web UI or any remote management, a management interface must be configured for chassis access. See [Configure an IP interface on the 10/100 BaseT Ethernet port for MXK out-of-band management on page 48](#).

For ZMS refer to *NetHorizon User's Guide*, *ZMS Administrator's Guide*, and the *ZMS Installation Guide*. For OSS Gateway, refer to OSS Gateway documentation.

Configure the MXK to run ZMS in SNMPv3

Configuring the MXK to run ZMS in SNMPv3

In order to invoke SNMPv3 for ZMS, you must delete ZMS, update **system 0**, and rerunning ZMS.

- 1 Delete the device connected to ZMS that is running SNMPv2.
- 2 Update the **system 0** file on the MXK with the *snmpv3includingZMS* variable for the **snmpVersion** parameter by deleting ZMS.

```
zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {false}:
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: ---> {false}:
configsyncpriority: ---> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: ---> {172.16.160.49_4_1392921484267}:
configsyncstatus: -----> {synccomplete}:
configsyncuser: -----> {zmsftp}:
configsyncpasswd: -----> {** private **}: ** read-only **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {172.16.160.49}:
alternateipaddress: ---> {0.0.0.0}:
countryregion: -----> {us}:
```

```

primaryclocksource: ---> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {disdefpktrules}:
reservedVlanIdStart: --> {0}:
reservedVlanIdCount: --> {0}:
snmpVersion: -----> {snmpv2}: snmpv3includingZMS
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

3 Reconnect the device to ZMS that is running SNMPv3.

In ZMS, open the region, select the correct region, then right-click **Add Device**.

From the **SNMP Version** drop-down menu in the **Add Device Configuration** dialog box, select **SNMP V3**.

Mass provisioning from the CLI when running ZMS

In order to perform mass provisioning from the CLI when ZMS is running, you must disable partial config sync traps to ZMS from the device. See [Configure an IP interface on the 10/100 BaseT Ethernet port for MXK out-of-band management on page 48](#).



Note: For how to enable ZMS, refer to the *NetHorizon User's Guide*.

CLI mass provisioning and ZMS

If you need to perform mass provisioning tasks with a script from the CLI when ZMS is managing the device, you must first disable ZMS in the **system 0** profile, complete the mass provisioning, enable ZMS again, and perform a config sync in ZMS.

1 Disable ZMS from managing the device, change the **zmsexists** parameter from *true* to *false*:

```

zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:

```



```

zmsexists: -----> {true}: false
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: ---> {false}:
configsyncpriority: ---> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: ---> {}:
configsyncstatus: -----> {syncinitializing}:
configsyncuser: -----> {}:
configsyncpasswd: -----> {** private **}: ** read-only **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {0.0.0.0}:
alternateipaddress: ---> {0.0.0.0}:
countryregion: -----> {us}:
primaryclocksource: ---> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {NONE(0)}:
reservedVlanIdStart: --> {0}
reservedVlanIdCount: --> {0}
snmpVersion: -----> {snmpv2}
persistentLogging: -----> {disabled}
outletTemperatureHighThreshold: -> {65}
outletTemperatureLowThreshold: --> {-12}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

2 Enable ZMS to manage the device, change the **zmsexists** parameter from *false* to *true*:

```

zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {true}: true
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: ---> {false}:
configsyncpriority: ---> {high}:

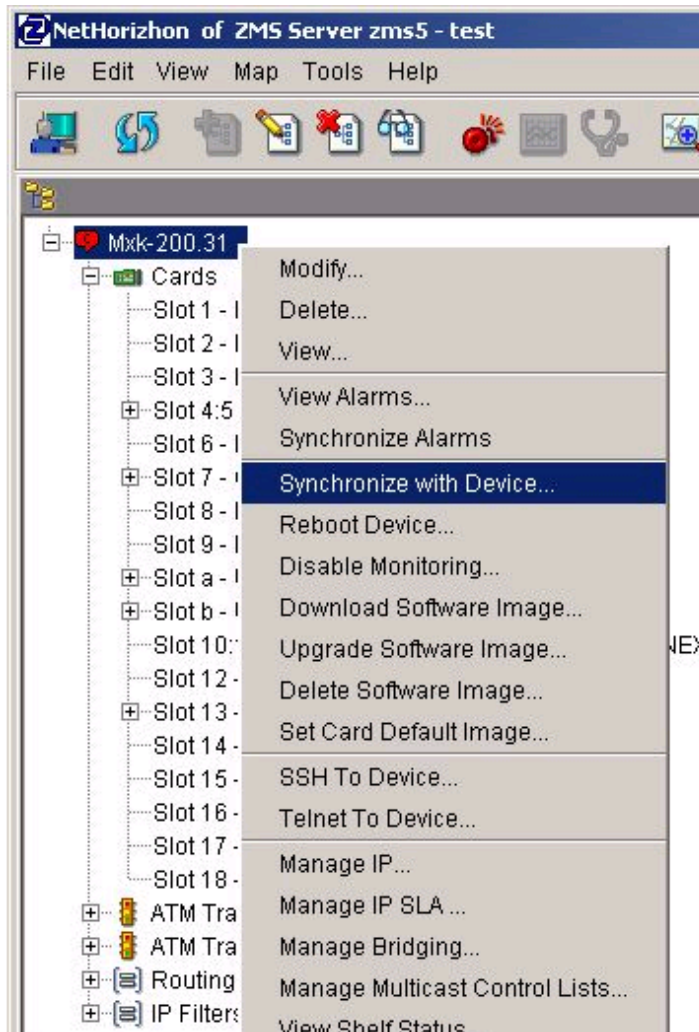
```

```

configsyncaction: ----> {noaction}:
configsyncfilename: ---> {}:
configsyncstatus: ----> {syncinitializing}:
configsyncuser: -----> {}:
configsyncpasswd: ----> {** private **}: ** read-only **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {0.0.0.0}:
alternateipaddress: ---> {0.0.0.0}:
countryregion: -----> {us}:
primaryclocksource: ---> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {NONE(0)}:
reservedVlanIdStart: --> {0}
reservedVlanIdCount: --> {0}
snmpVersion:-----> {snmpv2}
persistentLogging: ----> {disabled}
outletTemperatureHighThreshold: -> {65}
outletTemperatureLowThreshold: --> {-12}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

3 Perform a full config sync in ZMS.



Note: For details on using ZMS, refer to the *ZMS Administrator's Guide* and the *NetHorizon User's Guide*.

CLI configuration of a device being managed by the ZMS is disabled by default. Attempting to configure the device results in an error:

Manage the MXK from the WebUI

This section describes:

- [Manage the MXK using Zhone Web User Interface, page 64](#)
- [Disable the Web UI, page 65](#)

Before using Zhone Management System (ZMS), the Web UI or any remote management, the management interface must be configured. See [Configure an IP interface on the 10/100 BaseT Ethernet port for MXK out-of-band management on page 48](#).



Caution: If you are using a public and not a private IP address for the Web UI, to protect your management system, Zhone recommends that the port access profile is configured for the Telnet port (port 23) and the management subnet is specified. See [Port access security on page 132](#) for more information on setting up port security.

The MXK enables Web-based configuration using the Zhone SLMS Web Interface Tool. The Zhone SLMS Web Interface Tool supports configuration and management of both line and uplink cards.

Manage the MXK using Zhone Web User Interface

To manage the MXK using the Zhone Web User Interface (UI):

- Add an IP address to the interface to be used for management.

On the uplink cards, the interface on the 10/100 Ethernet port or GigE ports can be used. Ensure that the IP address is in the same subnet as the client devices and is reachable through Telnet. This example adds an IP interface for 172.24.94.103 to the 10/100 Ethernet port using VLAN 94.

```
zSH> interface add 1-a-1-0/eth vlan 94 172.24.94.103/24
Created ip-interface-record ethernet1-94/ip
```

- Configure a default route to the IP interface.

The default route enables connectivity to the IP interface.

```
zSH> route add default 94 172.24.94.103 metric 1
```

To launch the Zhone Web User Interface, in a browser URL address space on a PC with connectivity to the MALC, enter the IP address configured on the MXK.



Note: Ipv4 is required for all IP termination on the MXK, including management interfaces. IPv6 is not supported for IP termination on the MXK.

The Zhone Web User Interface launches and displays the Login window for the MXK.

Figure 4: Zhone Web User Interface Login Screen



MXK823

The Multi-Access Line Concentrator, MXK, platform provides low-cost, high-density subscriber access concentration in the Zhone Single Line Multi-Service (SLMS) architecture. The MXK is designed as a high-bandwidth, non-blocking Ethernet based access platform that integrates high bandwidth access technologies (GPON and active Ethernet) of core/metro Ethernet networks with the breadth and flexibility of interfaces required at the edge of networks.

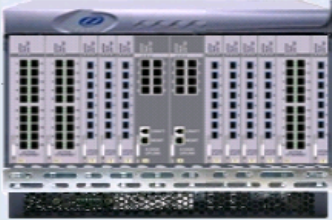
The MXK can be deployed in Central Office environments or controlled environmental vaults for remote terminal applications. The MXK is intended for restricted access locations only.

Web Interface Login

User Name

Password

MXK 823



On the Login page, enter the user name and password. The default user name is *admin* and the default password is *zhone*.



Note: Zhone recommends you change the user name and password to ones suitable to your network.

Click the desired menu to display the management options. For online help, click the Help icon  or product title in any window.

Disable the Web UI

Disabling the Web UI

Delete the *mxk823_http.tar* or *mxk819_http.tar* file from the *card1* directory to remove the software file associated with the Web UI. If you remove this file, you would have to reinstall the file in the *card1* directory to run the Web UI.

1 Verify the current directory.

```
zSH> pwd
/card1
```

2 Verify the *mxk823_http.tar* or *mxk819_http.tar* file in the *card1* directory.

```
zSH> dir
Listing Directory .:
-rwxrwxrwx 1 0      0      852028 Aug 22 11:51 mxup2tg8graw.bin
-rwxrwxrwx 1 0      0      13080567 Aug 22 11:51 mxup2tg8g.bin
-rwxrwxrwx 1 0      0      5310220 Sep  1 2011 mxlc48aads1bond.bin
-rwxrwxrwx 1 0      0      1100640 Jan 17 2011 malcmtac.bin
-rwxrwxrwx 1 0      0      1321250 Jan 17 2011 malcmtacenh.bin
-rwxrwxrwx 1 0      0      3788749 Jan 17 2011 mxlc48adsl2p.bin
-rwxrwxrwx 1 0      0      1322775 Jan 17 2011 tacitmring.bin
drwxrwxrwx 1 0      0      4096 Dec 21 2010 crash/
-rwxrwxrwx 1 0      0      4418987 Jan 17 2011 mxlcgp.bin
drwxrwxrwx 1 0      0      4096 Aug 22 13:35 datastor/
drwxrwxrwx 1 0      0      4096 Jan 17 2011 onreboot/
drwxrwxrwx 1 0      0      4096 Aug 22 13:34 log/
drwxrwxrwx 1 0      0      4096 Jul 27 2000 bulkstats/
drwxrwxrwx 1 0      0      4096 Jun  4 2010 pub/
-rwxrwxrwx 1 0      0      4257603 Sep  1 2011 mxlc24gshdslbond.bin
-rwxrwxrwx 1 0      0      5021611 Sep  1 2011 mxlc20ae.bin
-rwxrwxrwx 1 0      0      7341267 Aug 22 11:49 mxlc4gp.bin
drwxrwxrwx 1 0      0      4096 Jan 17 2011 me/
drwxrwxrwx 1 0      0      4096 Jan 17 2011 omci/
-rwxrwxrwx 1 0      0      405552 Jan 17 2011 mxlc20aerom.bin
-rwxrwxrwx 1 0      0      7341728 Aug 22 11:50 mxlc8gp.bin
-rwxrwxrwx 1 0      0      18428 Jan 17 2011 znid-gpon-2510-omci.txt
-rwxrwxrwx 1 0      0      9249280 Aug 22 11:48 mxk819_http.tar
-rwxrwxrwx 1 0      0      18428 Jan 17 2011 dumpme1
-rwxrwxrwx 1 0      0      748 Jan 17 2011 rsa.der
-rwxrwxrwx 1 0      0      1058 Jan 17 2011 rsakey.dat
drwxrwxrwx 1 0      0      4096 Jan 17 2011 newme/
drwxrwxrwx 1 0      0      4096 Jan 17 2011 1.16.2.123/
-rwxrwxrwx 1 0      0      9663488 Jan 17 2011 mxk823_http.tar
-rwxrwxrwx 1 0      0      5094732 Aug 22 11:48 mxlc20ae1s.bin
-rwxrwxrwx 1 0      0      7461652 Aug 22 11:49 mxlc24vds12.bin
-rwxrwxrwx 1 0      0      852028 Jan 17 2011 mxup8graw.bin
-rwxrwxrwx 1 0      0      5694994 Jan 17 2011 mxlc48badslbond.bin
147661088 bytes available
```

3 Delete the *mxk823_http.tar* file.

```
zSH> del mxk823_http.tar
```

The file is removed from the MXK. The file must be reinstalled in the *card1* directory to run the Web UI.

MXK system administration

This section describes how to work with profiles in the MXK system, and many of the CLI commands that are useful when performing system administration tasks and includes a discussion of:

- [MXK system defaults, page 67](#)
- [User account administration, page 68](#)
- [View chassis and system information, page 73](#)
- [View runtime statistics for the MXK with the card stats command, page 79](#)
- [Monitor the system with log files, page 81](#)
- [Navigate the MXK file system, page 93](#)
- [MXK basic system administration commands, page 96](#)
- [Save and restore configurations, page 105](#)
- [SNTP, page 106](#)
- [MXK Simple Network Management Protocol \(SNMP\), page 108](#)

MXK system defaults

This section describes the MXK system defaults, monitoring the MXK, and temporary logging sessions:

- [Defaults overview, page 67](#)
- [Monitoring the MXK through the serial craft port, page 68](#)
- [Enable/disable temporary logging sessions, page 68](#)

Defaults overview

The MXK must have at least one uplink card installed before the MXK will boot properly. Along with the ability to display cards (both active and inactive) which are in the MXK, you can also see into the DOS file system which stores boot code, software images, and configurations. See [Navigate the MXK file system on page 93](#) for a description of commands which can be used to access the MXK file system.

Line cards (except the first uplink card in slot *a*) must be provisioned with a card-profile before they will boot up.

- Administrative user name is **admin**, password is **zhone**.
- A single record for the Ethernet interface on the uplink card in slot *a* exists. No other profiles to configure physical interfaces exist.

- The uplink card in slot *a* is enabled. You must enable all other cards including the uplink card in slot *b* in a **card-profile** before they will boot up.
- A default **system 0** profile exists with the following configuration:
 - Authentication traps are not enabled
 - ZMS communication is not configured
 - Alarm notification and output are enabled for all severity levels

Monitoring the MXK through the serial craft port

The MXK can send messages to a console session, a log file, or to a syslog server and be configured to a number of system event levels — emergency, alert, critical, error, warning, notice, information, and debug.

Enable/disable temporary logging sessions

By default, log messages are enabled on the serial craft port. Use the **log session** command and the **log serial** command to enable/disable logging:

The **log session** command enables/disables logging messages for that session only when connected to the device through a Telnet session. If the user logs out, the logging setting returns to the default. To enable/disable logging for the current Telnet session only enter:

```
zSH> log session on
Logging enabled.
```

```
zSH> log session off
Logging disabled.
```

This command setting does not persist across system reboots.

The **log serial** command enables/disables logging messages for the session on the serial craft port. This command can be used in both Telnet connections and serial port connections to turn on and off the serial craft port logs. To enable/disable logging for the serial craft port enter:

```
zSH> log serial on
Serial port logging enabled.
```

```
zSH> log serial off
Serial port logging disabled.
```

This command setting persists across system reboots.

User account administration

MXK users have access to the CLI and are able to configure and administer the system.

- [Add users, page 69](#)
- [Create an SNMP v3 user from CLI, page 70](#)
- [Change default user passwords, page 70](#)
- [Delete users, page 70](#)
- [Delete the admin user account, page 71](#)
- [Reset passwords, page 71](#)

Add users

Every administrative user on the system must have a user account. The account specifies their username and password, as well as their privilege level, which determines their access to commands.

Users with **admin** privileges have access to all the administrative commands. Users with **user** privileges have access to a very limited set of commands. The highest level of access is **useradmin**, which allows the creation of user accounts.



Note: When entering access level responses, enter **yes** completely or the CLI interprets the response as **no**.

To add a user, enter the following commands:

```
zSH> adduser
Please provide the following: [q]uit.
User Name: jjsmith
User Prompt[zSH>]:

Please select user access levels.
admin: -----> {no}: yes
zhonedebug: --> {no}:
voice: -----> {no}:
data: -----> {no}:
manuf: -----> {no}:
database: ----> {no}:
systems: ----> {no}:
tool: -----> {no}:
useradmin: ---> {no}: yes
.....
User name:(jjsmith) User prompt:(zSH>)
Access Levels:
(admin)(useradmin)
Save new account? [s]ave, [c]hange or [q]uit: s
User record saved.
TEMPORARY PASSWORD: hmj4mxFU
```

Commands with **zhonedebug** privilege levels are intended for use by Zhone development only.

Immediately after activating the user account, you should change the password something you can remember, as explained in the next section.

Create an SNMP v3 user from CLI

Creating an SNMP v3 user

- 1 Use the **adduser snmp** *username* command to create an SNMPv3 user. Select the Auth protocol and the Priv protocol, then enter a password if prompted.

For example:

```
zSH> adduser snmp test
Auth protocol (md5, sha, or none): md5
Enter auth password:
Confirm auth password:
Priv Protocol (des or none): des
Enter priv password:
Confirm priv password:
Enter access group (readwrite, readonly, encrypt, admin) : readwrite
```

- 2 Verify the user.

```
zSH> showuser snmp
```

userName	auth	priv	accessGroup
-----	----	----	-----
zmsUser	md5	des	readwrite
test	md5	des	readwrite

Change default user passwords

When adding users, the system automatically assigns a temporary password to each user. Most users will want to change their password. The **changepass** command changes the password for the current logged in user. The following is an example of changing a password:

```
zSH> changepass
Current Password:
New Password:
Confirm New Password:
Password change successful.
```

Delete users

To delete a user, enter the **deleteuser** command and specify the username:

```
zSH> deleteuser jsmith
OK to delete this account? [yes] or [no]: yes
User record deleted.
```

Delete the admin user account

In addition to deleting regular user accounts, you can also delete the **admin** user account. This account is automatically created by the system and provides full access to the CLI.



Note: You cannot delete the **admin** account (or any other user account with **useradmin** privileges) if you are currently logged into it.

To delete the **admin** account:

```
zSH> deleteuser admin
```

If desired, you can recreate an account named **admin** after deleting it:

```
zSH> adduser admin
Please provide the following: [q]uit.
User Name: admin
User Prompt [zSH>]:

Please select user access levels.
admin: -----> {no}: yes
zhonedebug: --> {no}:
voice: -----> {no}: yes
data: -----> {no}: yes
manuf: -----> {no}: yes
database: -----> {no}: yes
systems: -----> {no}: yes
tool: -----> {no}: yes
useradmin: ---> {no}: yes
.....
User name: (admin) User prompt: (zSH>)
Access Levels:
(admin) (voice) (data) (manuf) (database) (systems) (tools) (useradmin)
Save new account? [s]ave, [c]hange or [q]uit: s
User record saved.
TEMPORARY PASSWORD: hmj4mxFU
```

Reset passwords

If a user forgets their password, an administrative user can reset the password and generate a new one using the **resetpass** command, as in the following example:

```
zSH> resetpass jsmith
Password:
```

user command

The **user** command enables the command line feature to add, modify, show, or delete users and user settings.

Syntax `user add <user-name> [password string] [prompt string] [admin] [debug] [voice] [data] [manuf] [dbase] [systems] [tools] [useradmin] [all]`

```
user modify <user-name> [password string] [prompt
string] [admin true|false] [debug true|false] [voice
true|false] [data true|false] [manuf true|false] [dbase
true|false] [systems true|false] [tools true|false]
[useradmin true|false] [all true|false]
```

changes user profile parameters

password option to set a new password

prompt option to set a new user prompt

other options set user access levels

"all" sets all access levels. It is processed before the other access level keywords, i.e. you can "manuf false all true".

That will set all access levels except manuf level access.

```
user delete <user-name>
```

deletes user account

```
user show [<user-name>]
```

displays user account information

displays all user accounts if no user-name entered

Options add

Adds a new user profile with the specified settings.

username

Name of the user.

password password

Specifies the password assigned to this user.

prompt

Specifies the system prompt to display for this user. If no password is entered, the system assigns a random password. Enclosing an argument in quotes allows the entry of special characters.

access level

Specifies the access levels assigned to the user. The **all** option sets all access levels. Individual access levels can be specified by added the keyword true or false after an access level. For example, **manuf false all true** sets all access levels except manuf level access.

Example 1

```
zSH> user add steve password pass prompt "zSH >" admin voice systems dbase
User record saved.
.....
User name:(Steve)   User prompt:(zSH >)
Access Levels:
(admin) (voice) (system) (dbase)
```

Example 2

```
zSH> user modify joe password pass all false admin true
OK to modify this account? [yes] or [no]: yes
User record updated.
.....
User name:(newaccount2)   User prompt:(zSH>)
Access Levels:
(admin) (useradmin)
```

Example 3

```
zSH> user show
.....
User name:(admin)   User prompt:(zSH>)
Access Levels:
(admin) (voice) (data) (manuf) (database) (systems) (tool) (useradmin)
.....
User name:(steve)   User prompt:(zSH>)
Access Levels:
(admin) (voice) (systems) (dbase)
.....
User name:(joe)   User prompt:(test >)
Access Levels:
(admin)
.....
User name:(kathy)   User prompt:(test4 >)
Access Levels:
(admin) (zhonedebug) (voice) (data) (manuf) (database) (systems) (tool) (useradmin)

zSH> user show steve
.....
User name:(steve)   User prompt:(zSH>)
Access Levels:
(admin) (voice) (systems) (dbase)
```

Example 4

```
zSH> user delete kathy
OK to delete this account? [yes] or [no]: yes
Account kathy deleted
```

View chassis and system information

This section describes:

- [MXK 819 and 823 fan tray monitoring, page 74](#)
- [MXK 319 fan tray monitoring, page 75](#)
- [MXK built-in alarm input output, page 77](#)

MXK 819 and 823 fan tray monitoring

The MXK supports monitoring the chassis/fan tray through the CLI.

The fan trays for the MXK 819 and MXK 823 support enhanced monitoring capabilities:

- individual fan rotation
- ambient air temperature
- three-point exhaust air temperature
- battery and return voltage measurement

To view overall status of the system, use the **shelfctrl monitor** command:

```
zSH> shelfctrl monitor
Shelf                               Status
-----
Uptime                             16 minutes
FPGA version                        0.5
Firmware version                    0.5
Uplink Supervisor CPLD version     1.3
Uplink Glue version                 0.2
16 MHz TDM clock                    Yes
Temperature Sensor                  Celsius (C)          Fahrenheit (F)
-----
Outlet sensor                       24              75
Temperature reading                  normal
Fan Power Supplies & Alarm           Status
-----
Fan Power A                         normal
Fan Power B                         normal
Fan alarm                           ok
Power Supplies                      Status
-----
Battery A                           normal
Battery B                           normal
Device                              Status
-----
System                              Critical alarm set
Card a                              Critical alarm set
Alarm I/O Board
-----
Supported:                          No
Present:                             No
```

System and **Card a** will show **Critical alarm set** when an alarm has been triggered. Other parameters provide full descriptions such as **warning fans A, B, C, F are stopped** or **warning all fans are stopped** for the Fan alarm.

The Battery A and Battery B voltages are measured relative to battery return (+). The Battery return voltage measurement is relative to ground (i.e., the chassis).

Note that earlier versions of the MXK 819/MXK 823 fan tray do not support all the monitoring functionality shown here. Consult your Zhone sales person for more information. See [MXK built-in alarm input output on page 77](#) for a description of the Alarm I/O Board functionality.

MXK 319 fan tray monitoring

The MXK 319 fan tray supports a subset of the monitoring features.

```
zSH> shelfctrl monitor
```

Shelf	Status

Uptime	4 days, 3 hours, 29 minutes
FPGA version	0.4
Firmware version	0.0
Temperature Sensor	Celsius (C)
Fahrenheit (F)	

Outlet sensor	35 95
Temperature reading	normal
Fan Power Supplies & Alarm	Status

Fan Power A	normal
Fan Power B	normal
Fan alarm	ok
Power Supplies	Status

Battery A	normal
Battery B	normal
Device	Status

System	
Card a	

To verify whether the shelf is active:

```
zSH> shelfctrl show
```

```
Shelf Controller Address: 01:a:12
Shelf Registry Address: 01:a:1042
Lease ID: 0x02070000_00000033
State: active
Slot 1:
    prevState: CONFIGURING currentState: RUNNING
    mode: FUNCTIONAL startTime: 1280841388
Slot 2:
    prevState: NONE currentState: NONE
    mode: NONE startTime: 0
Slot 3:
    prevState: NONE currentState: NONE
    mode: NONE startTime: 0
Slot 4:
    prevState: NONE currentState: NONE
    mode: NONE startTime: 0
Slot 5:
    prevState: CONFIGURING currentState: RUNNING
    mode: FUNCTIONAL startTime: 1280845212
Slot 6:
    prevState: CONFIGURING currentState: RUNNING
    mode: FUNCTIONAL startTime: 1280910574
Slot 7:
    prevState: CONFIGURING currentState: RUNNING
    mode: FUNCTIONAL startTime: 1280828845
Slot 8:
    prevState: CONFIGURING currentState: RUNNING
    mode: FUNCTIONAL startTime: 1280837242
Slot 9:
    prevState: NONE currentState: NONE
    mode: NONE startTime: 0
Slot 10:
    prevState: CONFIGURING currentState: RUNNING
    mode: FUNCTIONAL startTime: 1280838736
Slot 11:
    prevState: NONE currentState: NONE
    mode: NONE startTime: 0
Slot 12:
    prevState: CONFIGURING currentState: RUNNING
    mode: FUNCTIONAL startTime: 1280828805
Slot 13:
    prevState: NONE currentState: NONE
    mode: NONE startTime: 0
Slot 14:
    prevState: NONE currentState: NONE
    mode: NONE startTime: 0
Slot a:
    prevState: CONFIGURING currentState: RUNNING
    mode: FUNCTIONAL startTime: 1280828709
Slot b:
    prevState: NONE currentState: NONE
    mode: NONE startTime: 0
```

To view system statistics enter:


```
zSH> shelfctrl stats
Shelf Controller Message Statistics
-----
Directory services: 2
Clock: 275089
Lease: 1050
Heartbeat: 551264
Card status: 10
Info: 11
Card updates: 27
```

MXK built-in alarm input output

Because the POTS line cards have both integrated ringing power and line test capabilities, the TAC card is no longer an essential component of installations except for the need for alarm inputs and reference clock inputs. To remove the need for alarm inputs, the new version MXK chassis has an alarm board with both input and output relays.

The *MXK Hardware Installation Guide* shows the location and description of the alarm input and output relays.

With the 2.3 release, the **shelfctrl monitor** command will display an **Alarm I/O Board** section at the bottom of the display. Note: the display has been truncated to show the new section (highlighted in bold).

```
zSH> shelfctrl monitorShelf
Status
-----
-----
Uptime                               1 minute
FPGA version                         0.5
Firmware version                     0.6
Uplink Supervisor CPLD version      1.4
Uplink Glue version                  0.2
16 MHz TDM clock                     Yes
...
Device                               Status
-----
-----
System                               No alarms reported

Alarm I/O
Board-----
-----
Supported:                           Yes
Present:                             Yes
Alarm input:                         Ai1  Ai2  Ai3  Ai4  Ai5  Ai6  Ai7  Ai8
Status (Energized/de-energized):    d    d    d    d    d    d    d    d
dNormalOpen/NormalClosed/NotSpec:   NS   NS   NS   NS   NS   NS   NS   NS
Alarm Active:                        No   No   No   No   No   No   No   No
No   No   No
```

Older MXK chassis which do not have the Alarm I/O board running the 2.3 or newer software will show that the Alarm I/O board is not present (highlighted).

```

zSH> shelfctrl monitor
Shelf                                     Status
-----
Uptime                                  15 days, 23 hours, 34
minutes
FPGA version                           0.5
Firmware version                       0.5
Uplink Supervisor CPLD version         1.3
Uplink Glue version                   0.2
16 MHz TDM clock                       Yes
...
Device                                 Status
-----
SystemNo alarms reported
Card aNo alarms reported

Alarm I/O Board
-----
Supported:                        No
Present:                         No

```

To support the Alarm I/O board, the correct uplink card and firmware needs to be present. For the 4x1G uplinks, the firmware is automatically upgraded when the software is upgraded to 2.3 or later.

The 8x1G and 2x10G+8x1G uplink cards do not upgrade automatically. Some of these uplinks with upgraded firmware are already in the field. To determine which uplink you have, use the **shelfctrl monitor** command:

- If the **shelfctrl monitor** display for **Alarm I/O Board** shows **Supported: Yes**, then **Present: Yes** then the alarm I/O board is present.
- If the **shelfctrl monitor** display for **Alarm I/O Board** shows **Supported: Yes**, the firmware is upgraded.
- If the **Alarm I/O Board** shows **Supported: No**, the uplink card does not support the alarm I/O board. Contact Zhone support.

Adding a description to a chassis alarm

The num2str-profile uses an index in the form:

```
/slot/282/alarm-contact
```

For the new MXK I/O alarm board, shelf must be 1, slot must be 0.

For example, the following example adds a description in the name field, and specifies normallyclosed in the normal-state field to the sixth alarm contact of the MXK i/o alarm board.

```
zSH> update num2str-profile 1/0/282/6
```

Please provide the following: [q]uit.

name: -----> {Relay 6}: cabinet open

normal-state: -> {notspecified}: normallyclosed

.....

Save changes? [s]ave, [c]hange or [q]uit: s

Record updated.

The **normal-state** field has three value options: **notspecified**, **normallyclosed**, **normallyopen**.

View runtime statistics for the MXK with the card stats command

The **card stats** command displays runtime statistics for the MXK device.

```
zSH> card stats
```

----- cpu % utilization -----								----- memory (KB)-----				Card
Memory	uptime											
slot	idle	usage	high	services	framework	low	% Used	Total	Peak	Avail		
Status	ddd:hh:mm:ss			s/w version								
1	90	10	3	5	0	0	65.14	87227	56824	30410	1	-
OK	1:04:32:32			MXK 2.5.1.113								

The **card stats all** command displays information for all the cards.

```
zSH> card stats all
```

----- cpu % utilization -----								----- memory (KB)-----				Card
Memory	uptime											
slot	idle	usage	high	services	framework	low	% Used	Total	Peak	Avail		
Status	ddd:hh:mm:ss			s/w version								
2	97	3	1	0	0	3	34.71	100770	34987	65793	1	-
OK	6:22:11:51			MXK 2.5.1.113								
3	99	1	0	0	0	0	13.85	121685	16854	104832	1	-
OK	6:22:11:57			MXK 2.5.1.113								
4	92	8	4	2	0	0	40.05	104662	41923	62749	1	-
OK	6:22:11:10			MXK 2.5.1.113								
5	92	8	5	2	0	0	42.54	104596	44507	60100	1	-
OK	6:22:10:17			MXK 2.5.1.113								
6	92	8	6	1	0	2	34.01	109718	37320	72407	1	-
OK	6:22:12:29			MXK 2.5.1.113								
10	85	15	0	14	0	0	35.33	107438	38064	69476	1	-
OK	6:22:10:25			MXK 2.5.1.113								

```

a*   85   15   3   11   0   0   38.52 210359 81059 129334 1 -
OK           6:22:13:47 MXK 2.5.1.113

```

Table 3: card stats command fields

Section	Field
CPU % utilization	slot Textual description of the unit/card or access device type.
	idle Percentage of time the CPU has spent executing tasks with priority of 200 or less. Tasks with priority of 200 or less (the higher the number, the lower the priority) are considered idle tasks.
	usage Percentage of time the CPU has spent executing tasks with priority of 199 or higher
	high High priority tasks are primarily related to packet processing and critical system monitoring. Percentage of time the CPU has spent executing tasks with priority of 001 to 099. High priority tasks are primarily related to packet processing and critical system monitoring.
	services Services are primarily line monitoring tasks for line state and alarms. Percentage of time the CPU has spent executing tasks with priority of 100 to 179. Services tasks are primarily line monitoring tasks for line state and alarms.
	framework Framework tasks are primarily database and network management system related activities such as config synch and backup. Percentage of time the CPU has spent executing tasks with priority of 180 to 199. Framework tasks are primarily database and network management system related activities such as config synch and backup.
	low Percentage of time the CPU has spent executing tasks with priority of 200 to 250
memory (KB)	Used Percentage of time the CPU has spent executing tasks with priority of 199 or higher.
	Total The amount of physical memory contained by the device/card.

Table 3: card stats command fields (Continued)

Section	Field
	Peak The maximum physical memory that has been allocated at any time by the device/card.
	Avail The amount of physical memory that is unallocated and not in use by the device/card.
Card Memory Status	Memory status of the card sent with memory trap. A trap is sent when each condition occurs. 1 - ramMemOK less then 90% of ram is used 2 - ramMemLow more then 90% of ram is used 3 - flashMemOK enough flash for maximum database 4- flashMemLow not enough flash for maximum database 5 - flashMemOut no more flash memory, data no longer persistent
uptime ddd:hh:mm:ss	Uptime is calculated as sysUpTime - ifLastChange (assuming the device/card is running).
s/w version	Software version.

Monitor the system with log files

This section provides the following information on how logs work on the MXK

- [Overview, page 82](#)
- [Default log store level, page 82](#)
- [User login notification, page 82](#)
- [Enable/disable temporary logging sessions, page 68](#)
- [Log message format, page 83](#)
- [Modify logging levels, page 84](#)
- [Non-persistent log messages, page 86](#)
- [Persistent log messages, page 88](#)
- [Example log messages, page 88](#)
- [Log filter command, page 88](#)
- [Send messages to a syslog server, page 89](#)
- [Specify different log formats for system and syslog messages, page 91](#)

Overview

Logging enables administrators to monitor system events by generating system messages. It sends these messages to:

- A temporary management session (either on the serial craft port or over a Telnet session)
- Log modules to create permanent log files
- A syslog server (optional)

The type of information sent in these messages can be configured using the **log** command. By default, the system sends the same type of information to all log message destinations. If you want to send different types of messages to the syslog daemon, use the **syslog** command.

Default log store level

The default log store level is now set to emergency so by default the **log display** command displays only emergency level messages. Use the **log cache** command to display all messages that have been logged to console.

Use the **cd log** and **dir** commands to view the log file history. The log files in this directory record console activity on the MXK for the running image, and preserve a copy of the last two reboots. The files *consolelog1.txt* and *consolelog2.txt* hold 10000 lines of console output each. Once the file reaches 10000 lines, the filename is changed to *.old* and a new *.txt* file is used. After a reboot, the *.txt* files are also saved as *.old* files. Use the **consolelog display <filename>** command to view the contents for a consolelog file. These files are used for troubleshooting and system activity monitoring.

User login notification

Notifications of user login are sent to the console log.

```
zSH> MAR 11 17:28:20: alert : 1/a/1031: clitask1: User admin@172.16.48.232 logged in on slot a
```

Enable/disable logging

By default, log messages are enabled on the serial craft port. Use the **log session** command and the **log serial** command to enable/disable logging:

The **log session** command enables/disables logging messages for that session only. If the user logs out, the logging setting returns to the default. To enable logging for the current session only:

```
zSH> log session on
Logging enabled.
```

To disable logging for the session:

```
zSH> log session off
```

Logging disabled.

The **log serial** command enables/disables logging messages for all sessions on the serial craft port. This setting persists across system reboots. To enable/disable logging for the serial craft port:

```
zSH> log serial on
Serial port logging enabled.
```

To disable logging for the serial port:

```
zSH> log serial off
Serial port logging disabled.
```

Log message format

Log messages contain the following information:

Table 4: Default log message fields

Option	Description
Date	Date stamp of log message. Enabled by default.
Time	Time stamp of log message. Enabled by default.
Ticks	Current tick count. When the tick option is used, the date and time fields are not displayed.
Level	Logging level of the message. Enabled by default.
Address	The shelf and slot and application identifier causing the alarm.
Logtest	Log handle.
Taskname	Name of task that generated the log message. This is generally useful only for Zhone development engineers. Enabled by default.
Function	Function that generated the log message.
Line	Line in code that generated the log message. This is generally useful only for Zhone support staff.
Port	Port related to the log message.
Category	Category of the log message.
System	System related to the log message.
All	Controls all log message options.
Default	Controls the default log message options.
Message text	A description of the error that caused the alarm.

To change the information displayed in the log messages, use the **log option** command. First, display the available options:

```
zSH> log option
```

```

Usage: log option < time      | 1 > < on | off >
      < date      | 2 > < on | off >
      < level     | 3 > < on | off >
      < taskname  | 4 > < on | off >
      < taskid   | 5 > < on | off >
      < file      | 6 > < on | off >
      < function  | 7 > < on | off >
      < line      | 8 > < on | off >
      < port      | 9 > < on | off >
      < category  | 10 > < on | off >
      < system    | 11 > < on | off >
      < ticks     | 12 > < on | off >
      < stack     | 13 > < on | off >
      < globalticks | 14 > < on | off >
      < all       | 14 > < on | off >
      < default   | 15 > < on | off >
options 'time' & 'date' supercede option 'ticks'
time: date: level: address: log: port: category: system: (0x707)

```

Then, turn the option **on** or **off**. For example, the following command will turn the task ID on or off in log messages:

```

zSH> log option taskid on
time: date: level: address: log: taskid: port: category: system: (0x717)

zSH> log option taskid off
time: date: level: address: log: port: category: system: (0x707)

```

The following commands will turn on or off the tick count display in log messages:

```

zSH> log option ticks on
time: date: level: address: log: port: category: system: ticks: (0xf07)

zSH> log option ticks off
time: date: level: address: log: port: category: system: (0x707)

```

The following command will turn all options on in log messages:

```

zSH> log option all on
time: date: level: address: log: taskname: taskid: file: function: line: port:
category: system: ticks: stack: globalticks: (0x3fff)

```

Modify logging levels

To modify logging, use the **log** command. To modify syslog messages, use the **syslog** command.



Caution: Changing the log level may generate enough output to disrupt service.

To display the current levels for all logging modules, use the **log show** command:

```
zSH> log show
```

MODULE	LEVEL	STATUS
adslhdlr	error	enabled
adslprof	error	enabled
alarm_mgr	error	enabled
assert	error	enabled
atm_cc_mib_hdlr	error	enabled
atmmgragnt	error	enabled
bds	error	enabled
bds_client	error	enabled
bridge	error	enabled
bridgemib	error	enabled
bridgerp	error	enabled
bulkstats	error	enabled
bulkstatshdlr	error	enabled
cam	error	enabled
card	error	enabled
card_resource	error	enabled
carddeletehdlr	info	enabled
cardred	error	enabled
cardsvchdlr	error	enabled
ccrp	error	enabled
crrr	error	enabled
cesmibhdlr	error	enabled
cli	error	enabled
clkmgr	warning	enabled
....		

Logging levels determine the number of messages that are displayed on the console. The higher the log level, the more messages are displayed. The MXK supports the following log levels:

- 1: emergency
- 2: alert
- 3: critical
- 4: error
- 5: warning
- 6: notice
- 7: information
- 8: debug

To change the log level, use the **log module level** command. For example, the following command changes the card module logging level to emergency:



Caution: Changing the log level may generate enough output to disrupt service.

```
zSH> log level card emergency
Module: card at level: emergency
```

To enable or disable log levels for a module, use the log enable or log disable commands. For example:

```
zSH> log disable card
Module: card is now disabled
```

Non-persistent log messages

The **log cache** command displays the non-persistent log cache messages:

```
zSH> log cache
[1]: MAY 19 14:28:31: alert   : 1/a/1025: alarm_mgr: 01: a:06 Critical ETHERNET Down -
Ethernet line down
[2]: MAY 19 14:30:19: alert   : 1/13/1025: alarm_mgr: 01:13:01 Major ETHERNET Up -
Ethernet line up
[3]: MAY 19 14:32:12: alert   : 1/13/1025: alarm_mgr: 01:13:01 Major ETHERNET Down -
Ethernet line down
[4]: MAY 19 14:32:26: alert   : 1/13/1025: alarm_mgr: 01:13:02 Major ETHERNET Up -
Ethernet line up
[5]: MAY 19 14:33:27: alert   : 1/13/1025: alarm_mgr: 01:13:02 Major ETHERNET Down -
Ethernet line down
[6]: MAY 19 14:36:23: alert   : 1/4/1025: alarm_mgr: 01: 4:01:01 Minor ONU Down
Line 1/4/1/1/gpononu CAUSE: inactive
[7]: MAY 19 14:36:32: alert   : 1/4/1025: alarm_mgr: 01: 4:01:01 Minor ONU Up
Line 1/4/1/1/gpononu CAUSE: active
[8]: MAY 19 14:36:53: critical: 1/a/1035: rebootserver:
* * * * Slot Reboot : type = 2, shelf = 1, slot = 4
[9]: JAN 01 00:00:11: error   : 1/4/9   : tnettask: Unable to find ifnet pointer for
ifindex 0x2c0
[10]: JAN 01 00:00:11: error   : 1/4/9   : tnettask: Unable to find ifnet pointer for
ifindex 0x2c1
[11]: JAN 01 00:00:12: error   : 1/4/9   : tnettask: Unable to find ifnet pointer for
ifindex 0x2c2
[12]: MAY 19 14:40:32: notice  : 1/a/12   : shelfctrl: Card in slot 4 changed state to
RUNNING.
[14]: MAY 19 14:40:32: alert   : 1/4/1025: alarm_mgr: 01: 4:02 Critical OLT Up

Line 1/4/2/0/gponolt CAUSE: active
```

The **log cache max length** command sets the maximum number of log messages to store. The maximum log cache size is 2147483647, depending in the amount of memory available.

log cache max length

To change the current configured log cache size:

```
zSH> log cache max 200
```

Maximum number of log messages that can be saved: 200

The **log cache grep pattern** command searches through the log cache for the specified regular expression.

log cache grep pattern

The following example searches through the log cache for the string "Critical":

```
zSH> log cache grep Critical
```

Searching for: "Critical"

```
[1]: AUG 02 22:37:19: alert : 1/a/1025: alarm_mgr: 01: a:01 Critical ETHERNET Up -
Ethernet line up
```

```
[2]: AUG 02 22:37:34: alert : 1/a/1025: alarm_mgr: 01: a:02 Critical ETHERNET Down -
Ethernet uplink down
```

```
[3]: AUG 02 22:37:34: alert : 1/a/1025: alarm_mgr: 01: a:03 Critical ETHERNET Down -
Ethernet line down
```

The **log cache clear** command clears the log cache.

log cache clear

The **log cache size** command sets the maximum amount of memory for the log cache. Without options, displays the current log size.

```
zSH> log cache size
```

Number of log messages in the cache: 20

Total bytes used by the cache: 2052

The **log cache help** command displays the help information for the **log cache** command:

```
zSH> log cache help
```

```
Usage: log cache < max    > < length  >
        < grep   > < pattern >
        < clear  >
        < size   >
        < help   >
```

With no arguments the 'log cache' command prints out all the log messages currently in the cache.

The 'max' command is used to view/set the maximum number of log messages that can be cached at one time. If the cache is full then the oldest log is discarded and the new log is inserted. If no value is given then the current setting is displayed

The 'size' command is used to display the amount of memory currently being used by the log cache.

The 'clear' command is used to erase the log cache.

The 'grep' command is used for searching the log cache for a specific pattern. Extended regular expressions are supported.

Persistent log messages

Use the **log cache** command to view the persistent logs which only stores emergency level logs. For example:

```
zSH> log display
AUG 07 19:01:17: emergency: 1/a/12 : shelfctrl: Critical alarm set!
AUG 07 21:25:36: emergency: 1/a/12 : shelfctrl: Critical alarm set!
SEP 21 17:44:22: emergency: 1/a/12 : shelfctrl: Critical alarm set!
NOV 19 18:58:18: emergency: 1/a/12 : shelfctrl: Critical alarm set!
NOV 22 03:30:37: emergency: 1/a/12 : shelfctrl: Critical alarm set!
DEC 06 18:23:37: emergency: 1/a/12 : shelfctrl: Critical alarm set!
FEB 13 21:00:45: emergency: 1/a/12 : shelfctrl: Critical alarm set!
MAR 04 19:07:32: emergency: 1/a/12 : shelfctrl: Critical alarm set!
```

Example log messages

This section provides examples of how to interpret log messages.

The following message appears when a card in the MXK chassis comes up or goes down.

The most important parts of the message are the date and time the event occurred, the shelf/slot of the event, and the message text. The remainder of the information is only useful for Zhone development engineers.

For example:

```
MAR 11 17:46:20: alert : 1/6/1025: alarm_mgr: 01: 6:01
Minor ETHERNET Down - Ethernet line down
```

```
MAR 11 17:46:21: alert : 1/6/1025: alarm_mgr: 01: 6:01
Minor ETHERNET Up - Ethernet line up
```

```
MAR 11 17:48:30: alert : 1/5/1025: alarm_mgr: 01: 5:03
Critical OLT Up
Line 1/5/3/0/gponolt CAUSE: active
```

Log filter command

The **log filter** command is available as part of the log command functionality. This command enables users to show, set and delete log filters. Log filters limit the scope of log messages to a specific entity for troubleshooting and diagnostics. When a log filter is set, the filter is assigned an index number and only messages relate the specified entity are displayed. Filters can be set for an specific ifindex, slot/port, VCL, or subscriber.

log filter

Restrict the display of log messages to only the log messages for a specified entity.

Syntax `log filter show | set (ifindex|port slotport|vcl ifindex
vpi vci|subscriber endpoint) | delete`

```
zSH> log filter set ifindex 12
New filter saved.
```

```
zSH> log filter set port 5 24
New filter saved.
```

```
zSH> log filter set subscriber 22
New filter saved.
```

```
zSH> log filter show
```

Index	Type	Filter Parameters
1	Port	slot=1, port=1
2	Port	slot=1, port=4
3	IfIndex	IfIndex=12
4	Port	slot=5, port=24
6	IfIndex	IfIndex=100
7	IfIndex	IfIndex=104
8	IfIndex	IfIndex=109
9	IfIndex	IfIndex=103
10	IfIndex	IfIndex=107

```
zSH> log filter delete 10
Log filter 10 deleted
```

Send messages to a syslog server

[Table 5](#) describes the parameters in the **syslog-destination** profile you can modify to send messages to a syslog server.

Table 5: syslog-destination profile parameters

Parameter	Description
address	The IP address of the machine hosting the syslog server. Default: 0.0.0.0
port	The UDP port to which the syslog messages will be sent. Default: 514

Table 5: syslog-destination profile parameters (Continued)

Parameter	Description
facility	The syslog facility to which the syslog messages will be sent. Values: local0 local1 local2 local3 local4 local5 local6 local7 no-map Default: local0
severity	The severity level used to filter messages being set to the syslog server. Values: emergency alert critical error warning notice info debug Default: debug

```

zSH> new syslog-destination 1
Please provide the following: [q]uit.
address: --> {0.0.0.0}: 192.200.42.5 IP address of the syslog server
port: ----> {514}: leave at default
facility: -> {local0}:
severity: -> {debug}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```

Specify different log formats for system and syslog messages

Table 6 describes the **log-module** profile that supports the configuration of persistent log messages, syslog messages, and persistent storage levels by module. Modify this profile when you need to send different messages to admin sessions, the persistent logs, and the syslog server.

Table 6: log-module profile parameters

Parameter	Description
name	The name of the module whose logging is controlled by this profile. Default: logtest
display	Controls the display of messages on the system. Messages logged at this level and above will be displayed. Values: emergency alert critical error warning notice info debug Default: error

Table 6: log-module profile parameters (Continued)

Parameter	Description
syslog	Controls the format of messages sent to the syslog server described in the syslog-destination profile. This field is similar to the display field, except for the trackdisplay value. Values: emergency alert critical error warning notice info debug trackdisplay Messages logged at, and above, the level set in the display parameter will also be recorded in the syslog server. Default: trackdisplay
store	Controls the persistent storage of messages. This field is similar to the display field, except for the trackdisplay value. Values: emergency alert critical error warning notice info debug trackdisplay Messages logged at, and above, the level set in the display parameter will also be recorded in the syslog server. Default: trackdisplay

```

zSH> new log-module 1
Please provide the following: [q]uit.
name: ----> {logtest}: test1
display: -> {error}: warning
syslog: --> {trackdisplay}:
store: ---> {trackdisplay}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```

In this case, the **log-module 1** will display to the screen, all messages at and above **warning**. The variable **trackdisplay** means that the same messages as defined in **display** are also sent to the syslog and storage. If different level of

messages are needed for the different destinations, the variables for **display**, **syslog**, and **store** can be set at different levels.

Navigate the MXK file system

This section describes the MXK file system and includes:

- [Access the MXK file system, page 93](#)
- [Download software files, page 94](#)

Access the MXK file system

Use the following commands to access the MXK file system:

- **cd** Changes directory.
- **dir** Lists the contents of the directory.
- **pwd** Displays the current working directory.
- **image** Verifies software images and downloads software images on the flash to system memory.

The uplink card flash memory contains DOS file system that stores the system boot code, software images, and the configuration. During system startup, the software images on the flash are decompressed and loaded into memory.

Use the **cd**, **dir**, and **pwd** commands to list the contents of the file system, as in the following example:

Change directory.

```
zSH> cd /card1
```

Print the working directory.

```
zSH> pwd
/card1
```

List the directories in the current directory.

```
zSH> dir
Listing Directory .:
-rwxrwxrwx 1 0      0      852028 Aug 22 11:51 mxup2tg8graw.bin
-rwxrwxrwx 1 0      0     13080567 Aug 22 11:51 mxup2tg8g.bin
-rwxrwxrwx 1 0      0     5310220 Sep  1 2011 mxlc48aadslbond.bin
-rwxrwxrwx 1 0      0     1100640 Jan 17 2011 malcmtac.bin
-rwxrwxrwx 1 0      0     1321250 Jan 17 2011 malcmtacenh.bin
-rwxrwxrwx 1 0      0     3788749 Jan 17 2011 mxlc48adsl2p.bin
-rwxrwxrwx 1 0      0     1322775 Jan 17 2011 tacitmring.bin
drwxrwxrwx 1 0      0      4096 Dec 21 2010 crash/
-rwxrwxrwx 1 0      0     4418987 Jan 17 2011 mxlcgp.bin
drwxrwxrwx 1 0      0      4096 Aug 22 13:35 datastor/
drwxrwxrwx 1 0      0      4096 Jan 17 2011 onreboot/
drwxrwxrwx 1 0      0      4096 Aug 22 13:34 log/
```

```

drwxrwxrwx 1 0      0          4096 Jul 27 2000 bulkstats/
drwxrwxrwx 1 0      0          4096 Jun  4 2010 pub/
-rwxrwxrwx 1 0      0        4257603 Sep  1 2011 mxlc24gshdslbond.bin
-rwxrwxrwx 1 0      0       5021611 Sep  1 2011 mxlc20ae.bin
-rwxrwxrwx 1 0      0       7341267 Aug 22 11:49 mxlc4gp.bin
drwxrwxrwx 1 0      0          4096 Jan 17 2011 me/
drwxrwxrwx 1 0      0          4096 Jan 17 2011 omci/
-rwxrwxrwx 1 0      0       405552 Jan 17 2011 mxlc20aerom.bin
-rwxrwxrwx 1 0      0       7341728 Aug 22 11:50 mxlc8gp.bin
-rwxrwxrwx 1 0      0       18428 Jan 17 2011 znid-gpon-2510-omci.txt
-rwxrwxrwx 1 0      0       9249280 Aug 22 11:48 mxk819_http.tar
-rwxrwxrwx 1 0      0       18428 Jan 17 2011 dumpme1
-rwxrwxrwx 1 0      0        748 Jan 17 2011 rsa.der
-rwxrwxrwx 1 0      0       1058 Jan 17 2011 rsakey.dat
drwxrwxrwx 1 0      0          4096 Jan 17 2011 newme/
drwxrwxrwx 1 0      0          4096 Jan 17 2011 1.16.2.123/
-rwxrwxrwx 1 0      0      9663488 Jan 17 2011 mxk823_http.tar
-rwxrwxrwx 1 0      0      5094732 Aug 22 11:48 mxlc20ae1s.bin
-rwxrwxrwx 1 0      0      7461652 Aug 22 11:49 mxlc24vds12.bin
-rwxrwxrwx 1 0      0      852028 Jan 17 2011 mxup8graw.bin
-rwxrwxrwx 1 0      0     5694994 Jan 17 2011 mxlc48badslbond.bin
147661088 bytes available

```

Download software files

The MXK contains a TFTP client that enables you to download files from a network to the flash card file system using the **image** command. A software image for the uplink card and each type of line card must be downloaded.

The **image** command uses the following syntax:

```
image download tftphost imagefilename
```

The following example downloads the software image for the uplink card (*mxkup2tg8graw.bin*) from host 192.168.8.21 to the root directory of the first flash card:

```
image download 192.168.8.21 mxup2tg8graw.bin
```

Downloading software files

Download software files from the TFTP server to the MXK when you need to upgrade the system software:

- 1 Create the *onreboot* directory if one does not already exist and back up the current configuration file to a file named *restore*, then **cd** back to the root directory.

```

zSH> mkdir onreboot
zSH> cd onreboot
zSH> dump file restore
zSH> cd ..

```

The *restore* file is used to restore the system configuration or revert to a previous release, if desired. See [Step 5](#).

- 2 Copy the new system boot image software to the flash memory using the **image download** command.

```
zSH> image download 192.168.8.21 mxup2tg8g.bin
```

where 192.168.8.21 is the TFTP server, and *mxup2tg8g.bin* is the name of the software image.



Caution: Be sure to download the correct software for the system.

- 3 Initialize the flash card's boot partition with the new image on both the primary and standby uplink card (if present).

For a single uplink card enter:

```
zSH> image flash mxup2tg8g.bin 1 1
```

For redundant uplink cards enter:

```
zSH> image flash mxup2tg8g.bin 1 all
```

- 4 The **image** command can also verify image files on the flash card. It reads the contents of the file, verifies the file header, and verifies the file checksum. For example:

```
zSH> image verify mxup2tg8g.bin
File: mxup2tg8graw.bin
Size: 688320 bytes
Header Version: 1
Card Type: MX TWO TENGIGE EIGHT GIGE
Load Type: binary
Load Address: 0x00010000
Checksum: 0x2f66bb70
Image verify successful
```

The command reports any errors it finds in the file. Note that files are also verified as part of the download process.

- 5 Reset the system and restore the system configuration with the **systemreboot** command:

```
zSH> systemreboot
A restore file (/card1/onreboot/restore) is present.
A system reboot will result in a database restore.
Continue? (yes or no) [no]: yes
Do you want to reboot the system? (yes or no) [no] yes
Do you want to exit from this request? (yes or no) [yes] no
Are you sure? (yes or no) [no] yes
```

As shown above, when the *restore* file is present, the system displays

```
A restore file (/card1/onreboot/restore) is present.
```

and uses that file to restore the saved configuration to the MXK system.

After upgrading the software, the system automatically upgrades the software database to the new level.

MXK basic system administration commands

Commands: new, list, show, get, update, delete

This section describes these commands:

- [new command, page 96](#)
- [list command, page 96](#)
- [show command, page 99](#)
- [get command, page 101](#)
- [update command, page 102](#)
- [delete command, page 102](#)

new command

The **new** command can create new GPON traffic profiles.

```
zSH> new gpon-traffic-profile 1
gpon-traffic-profile 1
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}:
traffic-class: -----> {ubr}:
compensated: -----> {false}:
shared: -----> {false}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

list command

The **list** command displays all the profiles available on the MXK (partial list shown):

```
zSH> list
adsl-co-profile: shelf/slot/port
adsl-cpe-profile: shelf/slot/port
adsl-profile: shelf/slot/port
alarm-config: ifIndex
analog-fxo-cfg-profile: ifIndex
analog-fxs-cfg-profile: ifIndex
analog-if-cfg-profile: ifIndex
atm-cc: atmVcCrossConnectIndex
atm-if: ifIndex
atm-if-stats: ifIndex
atm-traf-descr: index
atm-traf-descr-stats: index
atm-vcl: ifIndex/vpi/vci
```

```

atm-vcl-param: index
atm-vcl-stats: ifIndex/vpi/vci
atm-vpi: ifIndex/vpi
atm-vpl: ifIndex/vpi
bridge-interface-record: ifIndex
bulk-statistic: index

```

The **list gpon-traffic-profile** command lists all GPON traffic profiles on the system.

```

zSH> list gpon-traffic-profile
gpon-traffic-profile 1
gpon-traffic-profile 2
gpon-traffic-profile 3
3 entries found.

```

The **list system** command displays the list of system profiles.

```

zSH> show system
syscontact:-----> {260}
sysname:-----> {260}
syslocation:-----> {260}
enableauthtraps:-----> enabled disabled
setserialno:-----> {0 - 2147483647}
zmsexists:-----> true false
zmsconnectionstatus:--> active inactive
zmsipaddress:-----> {0 - 0}
configsyncexists:-----> true false
configsyncoverflow:---> true false
configsyncpriority:---> none low medium high
configsyncaction:-----> noaction createlist createfulllist
configsyncfilename:---> {68}
configsyncstatus:-----> synccomplete syncpending syncerror syncinitializing
configsyncuser:-----> {36}
configsyncpasswd:-----> {36}
numshelves:-----> {0 - 0}
shelvesarray:-----> {36}
numcards:-----> {0 - 0}
ipaddress:-----> {0 - 0}
alternateipaddress:---> {0 - 0}
countryregion:-----> argentina australia belgium china costarica finland
france germany hongkong italy japan korea mexico netherlands newzealand
singapore spain sweden switzerland uk us afghanistan albania algeria
americansamoa andorra angola anguilla antarctica antiguabarbuda armenia aruba
austria azerbaijan bahamas bahrain bangladesh barbados belarus belize benin
bermuda bhutan bolivia bosniaherzegovina botswana bouvetisland brazil
britishindianoceanterritory bruneidarussalam bulgaria burkinafaso burundi
cambodia cameroon canada capeverde caymanislands centralafricanrepublic chad
chile christmasisland cocosislands colombia comoros congo cookislands
cotedivoire croatia cuba cyprus czechrepublic denmark djibouti dominica
dominicanrepublic easttimor ecuador egypt elsalvador equatorialguinea eritrea
estonia ethiopia falklandislands faroeislands fiji frenchguiana frenchpolynesia
frenchsouthernterritories gabon gambia georgia ghana gibraltar greece greenland
grenada guadeloupe guam guatemala guinea guineabissau guyana haiti

```

```

heardislandmcdonaldislands holysee honduras hungary iceland india indonesia
iran iraq ireland israel jamaica jordan kazakstan kenya kiribati northkorea
kuwait kyrgyzstan lao latvia lebanon lesotho liberia libyanarabjamahiriya
liechtenstein lithuania luxembourg macau macedonia madagascar malawi malaysia
maldives mali malta marshallislands martinique mauritania mauritius mayotte
micronesia moldova monaco mongolia montserrat morocco mozambique myanmar
namibia nauru nepal netherlandsantilles newcaledonia nicaragua niger nigeria
niue norfolkisland northernmarianaislands norway oman pakistan palau
palestinianterritory panama papuanewguinea paraguay peru philippines pitcairn
poland portugal puertorico qatar reunion romania russia rwanda sainthelena
saintkittsnevis saintlucia saintpierremiquelon saintvincentthegrenadines samoa
sanmarino saotomeprincipe saudiarabia senegal seychelles sierraleone slovakia
slovenia solomonislands somalia southafrica southgeorgia srilanka sudan
suriname svalbardjanmayen swaziland syria taiwan tajikistan tanzania thailand
togo tokelau tonga trinidadtobago tunisia turkey turkmenistan
turkscaicosislands uganda ukraine unitedarabemirates uruguay uzbekistan vanuatu
venezuela vietnam virginislandsuk virginislandsus wallisfutuna westernsahara
yemen yugoslavia zambia zimbabwe
primaryclocksource:--> [Shelf {0-255}/Slot {0-31}/Port {0-500}/SubPort/Type] |
[Name/Type]
ringsource:-----> internalringsourcelabel externalringsourcelabel
revertiveclocksource:--> true false
voicebandwidthcheck:--> true false
alarm-levels-enabled:--> critical+major+minor+warning
userauthmode:-----> local radius radiusthenlocal radiusthencraft
radiusauthindex:-----> {0 - 2147483647}
secure:-----> enabled disabled
webinterface:-----> enabled disabled
options:----->
cvlanonly+nol3bridgetable+ipg88bits+disdefpktrules+enablexcardlinkagg+fiberlan+cfmon
reservedVlanIdStart:--> {0 - 4090}
reservedVlanIdCount:--> {0 - 2048}
snmpVersion:-----> snmpv2 snmpv3 snmpv3includingZMS
persistentLogging:----> enabled disabled
outletTemperatureHighThreshold:--> {35 - 65}
outletTemperatureLowThreshold:--> {-40 - 0}

```

To view the card profiles existing on the system, enter **list card-profile**:

```

zSH> list card-profile
card-profile 1/a/10130
card-profile 1/b/10130
card-profile 1/1/10208
card-profile 1/3/10202
card-profile 1/5/10202
card-profile 1/10/10216
card-profile 1/11/10200
card-profile 1/13/10202
8 entries found.

```

To view the **bridge-interface-record** profiles of existing bridges enter **list bridge-interface-record**:

```

zSH> list bridge-interface-record

```

```
bridge-interface-record ethernet2-94/bridge
bridge-interface-record 1-1-1-0-eth-94/bridge
bridge-interface-record ethernet2-220/bridge
bridge-interface-record 1-1-1-0-eth-220/bridge
bridge-interface-record ethernet2-998/bridge
bridge-interface-record 1-1-1-0-eth-998/bridge
6 entries found.
```

show command

Use the **show** command to view all the options in a profile. For example, if you need to find which country codes are available on the MXK, use the **show system** command.

```
zSH> show system
syscontact:-----> {260}
sysname:-----> {260}
syslocation:-----> {260}
enableauthtraps:-----> enabled disabled
setserialno:-----> {0 - 2147483647}
zmsexists:-----> true false
zmsconnectionstatus:--> active inactive
zmsipaddress:-----> {0 - 0}
configsyncexists:-----> true false
configsyncoverflow:---> true false
configsyncpriority:---> none low medium high
configsyncaction:-----> noaction createlist createfulllist
configsyncfilename:---> {68}
configsyncstatus:-----> synccomplete syncpending syncerror syncinitializing
configsyncuser:-----> {36}
configsyncpasswd:-----> {36}
numshelves:-----> {0 - 0}
shelvesarray:-----> {36}
numcards:-----> {0 - 0}
ipaddress:-----> {0 - 0}
alternateipaddress:---> {0 - 0}
countryregion:-----> argentina australia belgium china costarica finland
france germany hongkong italy japan korea mexico netherlands newzealand
singapore spain sweden switzerland uk us afghanistan albania algeria
americansamoa andorra angola anguilla antarctica antiguabarbuda armenia aruba
austria azerbaijan bahamas bahrain bangladesh barbados belarus belize benin
bermuda bhutan bolivia bosniaherzegovina botswana bouvetisland brazil
britishindianoceanterritory bruneidarussalam bulgaria burkinafaso burundi
cambodia cameroon canada capeverde caymanislands centralafricanrepublic chad
chile christmasisland cocosislands colombia comoros congo cookislands
cotedivoire croatia cuba cyprus czechrepublic denmark djibouti dominica
dominicanrepublic easttimor ecuador egypt elsalvador equatorialguinea eritrea
estonia ethiopia falklandislands faroeislands fiji frenchguiana frenchpolynesia
frenchsouthernterritories gabon gambia georgia ghana gibraltar greece greenland
grenada guadeloupe guam guatemala guinea guineabissau guyana haiti
heardislandmcdonaldislands holysee honduras hungary iceland india indonesia
iran iraq ireland israel jamaica jordan kazakistan kenya kiribati northkorea
kuwait kyrgyzstan lao latvia lebanon lesotho liberia libyanarabjamahiriya
liechtenstein lithuania luxembourg macau macedonia madagascar malawi malaysia
```

```

maldives mali malta marshallislands martinique mauritania mauritius mayotte
micronesia moldova monaco mongolia montserrat morocco mozambique myanmar
namibia nauru nepal netherlandsantilles newcaledonia nicaragua niger nigeria
niue norfolkisland northernmarianaislands norway oman pakistan palau
palestinianterritory panama papuanewguinea paraguay peru philippines pitcairn
poland portugal puertorico qatar reunion romania russia rwanda sainthelena
saintkittsnevis saintlucia saintpierremiquelon saintvincentthegrenadines samoa
sanmarino saotomeprincipe saudiarabia senegal seychelles sierraleone slovakia
slovenia solomonislands somalia southafrica southgeorgia srilanka sudan
suriname svalbardjanmayen swaziland syria taiwan tajikistan tanzania thailand
togo tokelau tonga trinidadtobago tunisia turkey turkmenistan
turkscaicosislands uganda ukraine unitedarabemirates uruguay uzbekistan vanuatu
venezuela vietnam virginislandsuk virginislandsus wallisfutuna westernsahara
yemen yugoslavia zambia zimbabwe
primaryclocksource:--> [Shelf {0-255}/Slot {0-31}/Port {0-500}/SubPort/Type] |
[Name/Type]
ringsource:-----> internalringsourcelabel externalringsourcelabel
revertiveclocksource:--> true false
voicebandwidthcheck:--> true false
alarm-levels-enabled:--> critical+major+minor+warning
userauthmode:-----> local radius radiusthenlocal radiusthencraft
radiusauthindex:-----> {0 - 2147483647}
secure:-----> enabled disabled
webinterface:-----> enabled disabled
options:----->
cvlanonly+nol3bridgetable+ipg88bits+disdefpktrules+enablexcardlinkagg+fiberlan+cfmon
reservedVlanIdStart:--> {0 - 4090}
reservedVlanIdCount:--> {0 - 2048}
snmpVersion:-----> snmpv2 snmpv3 snmpv3includingZMS
persistentLogging:----> enabled disabled
outletTemperatureHighThreshold:--> {35 - 65}
outletTemperatureLowThreshold:--> {-40 - 0}

```

Use additional show commands such as **show bridge-interface-record** to view greater detail about bridges.

```

zSH> show bridge-interface-record
vpi:-----> {0 - 4095}
vci:-----> {0 - 65535}
vlanId:-----> {0 - 4090}
stripAndInsert:-----> false true
customARP:-----> false true
filterBroadcast:-----> false true
learnIp:-----> false true
learnUnicast:-----> false true
maxUnicast:-----> {0 - 2147483647}
learnMulticast:-----> false true
forwardToUnicast:-----> false true
forwardToMulticast:-----> false true
forwardToDefault:-----> false true
bridgeIfCustomDHCP:-----> false true
bridgeIfIngressPacketRuleGroupIndex:--> {0 - 2147483647}
vlanIdCOS:-----> {0 - 7}
outgoingCOSOption:-----> disable all

```



```

outgoingCOSValue:-----> {0 - 7}
s-tagTPID:-----> {33024 - 37376}
s-tagId:-----> {0 - 4090}
s-tagStripAndInsert:-----> false true
s-tagOutgoingCOSOption:-----> s-tagdisable s-tagall
s-tagIdCOS:-----> {0 - 7}
s-tagOutgoingCOSValue:-----> {0 - 7}
mcastControlList:-----> {264}
maxVideoStreams:-----> {0 - 1024}
isPPPoA:-----> false true
floodUnknown:-----> false true
floodMulticast:-----> false true
bridgeIfEgressPacketRuleGroupIndex:--> {0 - 2147483647}
bridgeIfTableBasedFilter:-----> none+mac+ip
bridgeIfDhcpLearn:-----> none+mac+ip
mvrVlan:-----> {0 - 4090}
vlan-xlate-from:-----> {0 - 4095}
slan-xlate-from:-----> {0 - 4095}
bridge-type:-----> uplink downlink intralink tls rlink
pppoa wire mvr user downlinkvideo downlinkdata downlinkpppoe downlinkp2p
downlinkvoice downlinkupstreammcast ipobtls ipobuplink ipobdownlink

```

get command

Use the **get** command to view the current configuration of profiles. The **get system 0** command displays information on the current MXK system configuration.

```

zSH> get system 0
system 0
syscontact: -----> {}
sysname: -----> {}
syslocation: -----> {}
enableauthtraps: -----> {disabled}
setserialno: -----> {0}
zmsexists: -----> {false}
zmsconnectionstatus: --> {inactive}
zmsipaddress: -----> {0.0.0.0}
configsyncexists: -----> {false}
configsyncoverflow: --> {false}
configsyncpriority: --> {high}
configsyncaction: -----> {noaction}
configsyncfilename: --> {172.16.160.49_4_1392921484267}
configsyncstatus: -----> {synccomplete}
configsyncuser: -----> {zmsftp}
configsyncpasswd: -----> ** private **
numshelves: -----> {1}
shelvesarray: -----> {}
numcards: -----> {3}
ipaddress: -----> {172.16.160.49}
alternateipaddress: --> {0.0.0.0}
countryregion: -----> {us}
primaryclocksource: --> {0/0/0/0/0}
ringsource: -----> {internalringsourcelabel}

```

```

revertiveclocksource: -> {true}
voicebandwidthcheck: --> {false}
alarm-levels-enabled: -> {critical+major+minor+warning}
userauthmode: -----> {local}
radiusauthindex: -----> {0}
secure: -----> {disabled}
webinterface: -----> {enabled}
options: -----> {disdefpktrules}
reservedVlanIdStart: --> {0}
reservedVlanIdCount: --> {0}
snmpVersion: -----> {snmpv2}
persistentLogging: ----> {disabled}
outletTemperatureHighThreshold: -> {65}
outletTemperatureLowThreshold: --> {-12}

```

You can find the syscontact information, or whether the MXK is configured to communicate with the Zhone Management System (ZMS — `zmsexists`, `zmsconnectionstatus`, `zmsipaddress`).

update command

To update the **system 0** profile and all other profiles, use the **update** command. The **update system 0** command walks you through the profile to change specific fields.



Caution: You should be very careful when altering profiles. Where available you should use CLI macro commands.

For example:

```

zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {true}: false
...
...
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

delete command

Use the **delete** command to delete profiles.

```

zSH> delete gpon-traffic-profile 1
gpon-traffic-profile 1
1 entry found.
Delete gpon-traffic-profile 1? [y]es, [n]o, [q]uit : y
gpon-traffic-profile 1 deleted.

```

Commands: interface show, bridge show

This section describes these commands:

- [interface show command, page 103](#)
- [bridge show command, page 103](#)

interface show command

The **interface show** command displays the numbered or unnumbered (floating) IP interfaces currently available on the MXK.

```
zSH> interface show
1 interface
Interface      Status  Rd/Address      Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 172.24.64.91/24  00:01:47:17:da:0e   ethernet1
-----
```

Table 7: Interface show column

Column	Description
Interface	Shows the interface, the card and the physical port of the IP interface.
Status	Shows whether the interface is up or down.
Rd/Address	The IP address assigned to this gateway.
Media/Dest Address	Media/Dest Address is either the MAC address of a device.
IfName	The interface name.

bridge show command

The **bridge show** command displays the bridge interfaces on the MXK. Note that a bridge is a combination of bridge interfaces working together.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
tls              3105      1/a/5/0/eth   ethernet5/bridge  UP  D 08:00:20:da:77:9c
                                     D 00:e0:39:ca:04:8e
                                     D 00:e0:39:98:97:2c
                                     D 00:60:e0:45:a9:ff
                                     D 00:50:04:bf:63:48
                                     D 00:30:48:2e:c8:f2
                                     D 00:30:19:81:b0:38
                                     D 00:08:9b:46:9b:26
                                     D 00:03:e3:97:bb:05
                                     D 00:03:e3:97:bb:00
                                     D 00:02:4b:74:d9:e2
                                     D 00:01:47:5c:34:58
                                     D 00:01:47:56:75:8e
                                     D 00:01:47:4e:dc:c0
                                     D 00:01:47:1a:e4:74
                                     D 00:01:47:14:c3:00
ipobtls      Tagged 3105      1/a/6/0/ipobridge  ipobridge-3105/bridge  UP  S 00:01:47:11:b7:c6
                                     S 10.51.5.5
```

2 Bridge Interfaces displayed

Use the **bridge show** command with a VLAN ID to view all the bridges on a VLAN.

```
zSH> bridge show vlan 999
Orig
Type VLAN/SLAN  VLAN/SLAN      Physical      Bridge      St
Table Data
-----
upl              Tagged  999    1/a/3/0/eth    ethernet3-999/bridge
UP  S VLAN 999 default
1 Bridge Interfaces displayed
```

Use the **bridge show <bridge interface>** command to view bridge interface information.

```
zSH> bridge show 1/7/3/16/gpononu
Orig
Type VLAN/SLAN  VLAN/SLAN      Physical      Bridge      St
Table Data
-----
dwn              Tg   101/502    1/7/3/16/gpononu    1-7-3-516-gponport-101/bridge
UP  D 00:00:ff:00:00:10
dwn              Tg   102/503    1/7/3/16/gpononu    1-7-3-516-gponport-102/bridge
UP
dwn              Tagged  500    1/7/3/16/gpononu    1-7-3-516-gponport-500/bridge
UP
tls              Tagged  848    1/7/3/16/gpononu    1-7-3-516-gponport-848/bridge
UP
dwn              Tagged  998    1/7/3/16/gpononu    1-7-3-916-gponport-998/bridge
UP  D 00:21:a1:aa:cd:10
tls              Tagged  2001   1/7/3/16/gpononu    1-7-3-516-gponport-2001/bridge
UP
6 Bridge Interfaces displayed
```

Commands: bridge stats

You can use the **bridge stats** command to view the packets being sent or received on bridge interfaces. If you add the name of a bridge you can see the statistics for that bridge.

```
zSH> bridge stats
Interface Counters
Name
Transmitted
ipobridge-3105/bridge
ethernet5/bridge
2 Bridge Interfaces displayed
```

	Received Packets				Transmitted Packets				Storm Detect Packets				Byte
	UCast	MCast	BCast	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	
ipobridge-3105/bridge	0	0	18	1	16	262	0	0	0	0	0	--	
ethernet5/bridge	--	--	--	--	--	--	--	0	0	0	0	--	

Commands: port show, port up, port down, port bounce, port status

You can use the **port** command to view the administrative state of an interface, change the administrative state of an interface, or change configuration parameters for an interface. See [MXK port management on page 111](#) for more information.

Enter **port show** <interface> to view the administrative state of an interface:

```
zSH> port show 1-6-2-0/eth
Interface 1-6-2-0/eth
  Physical location:      1/6/2/0/eth
  Administrative status: up
  Port type specific information:
    Frame size: 0 bytes
    Ingress rate: 0 Kbps burst size: 0 Kbits
    Egress rate: 0 Kbps burst size: 0 Kbits
```

Use port up, down, or bounce to alter the administrative status of a physical or virtual interface. Bounce performs a down operation followed by an up operation.

Enter **port up** <interface> to change the administrative state of an interface from down to up:

```
zSH> port up 1-6-2-0/eth
1-6-2-0/eth set to admin state UP
```

Enter **port down** <interface> to change the administrative state of an interface from up to down:

```
zSH> port down 1-6-2-0/eth
1-6-2-0/eth set to admin state DOWN
```

Enter **port bounce** <interface> to change the interface from UP to DOWN, and back to UP.

```
zSH> port bounce 1-6-2-0/eth
1-6-2-0/eth set to admin state DOWN
1-6-2-0/eth set to admin state UP
```

Enter the **port status** <interface> to get the operational status, speed and duplex mode of the Ethernet port.

```
zSH> port status 1-a-1-0/eth
Operational status : Up
Rate in Mbps       : 100
Duplex             : Full
```

Save and restore configurations

The **dump** command saves the system configuration to the console, a local file, or the network.

The command uses the following syntax:

```
dump [file filename] [network host filename]
```

Passwords are encrypted when they are saved to the configuration file. The encrypted passwords are used to restore the correct password, but cannot be used to log in.



Note: The **dump** command uses TFTP to transfer files to the network. Set the TFTP server time-out value to at least 5 seconds, and 5 retries to help prevent TFTP timeout or retry errors.

Backing up the configuration to a local file

To dump the configuration to a local file:

Specify a file name for the configuration:

```
zSH> dump file filename
```

The file is saved on the MXK file system.

Backing up the configuration to the network

To back up the configuration to the network:

- 1 Create the file in the destination location of the TFTP server and make it writeable.
- 2 Save the configuration. The following example saves the configuration to a file named `device.cfg` on the host `192.168.8.21`:

```
zSH> dump network 192.168.8.21 device.cfg
```

Restoring the configuration

The configuration is restored to the system during **systemreboot**. See [Downloading software files on page 94](#).

SNTP

Set system for SNTP

To set up the system to use SNTP update the **ntp-client-config** profile:

```
zSH> update ntp-client-config 0
ntp-client-config 0
Please provide the following: [q]uit.
primary-ntp-server-ip-address: ---> {0.0.0.0}: 192.168.8.100
secondary-ntp-server-ip-address: -> {0.0.0.0}:
local-timezone: -----> {gmt}: pacific
daylight-savings-time: -----> {false}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
```

Record updated.

Set Daylight Savings Time begin and end times

To set the specific date and time for the beginning and end of daylight savings time add the month, day and time in the **daylight-savings-time-start** and **daylight-savings-time-end** parameters of the **ntp-client-config** profile. Follow the MM:DD:HH:MM (month:day:hour:minute) format.

For example to set the daylight savings time to begin on March 10 at 2am and end on November 3 at 2am, the actual times for 2013 DST, you would update the **ntp-client-config** as shown below.

```
zSH> update ntp-client-config 0

ntp-client-config 0
Please provide the following: [q]uit.
primary-ntp-server-ip-address: ---> {172.16.1.53}:
secondary-ntp-server-ip-address: -> {0.0.0.0}:
local-timezone: -----> {pacific}:
daylight-savings-time: -----> {true}:
daylight-savings-time-start: -----> {03:10:02:00}:
daylight-savings-time-end: -----> {11:03:02:00}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```



Note: The **primary-ntp-server-ip-address** parameter must be non-zero to save changes to the **ntp-client-config** profile.



Note: When testing this feature, please ensure that there is at least two hours time between the start and end times of the cycle for the feature to operate correctly.

MXK Simple Network Management Protocol (SNMP)

This section describes the following:

- [Create SNMP community names and access profiles, page 108](#)
- [Configure traps, page 109](#)

Create SNMP community names and access profiles



Note: By default, the MXK has a single SNMP community defined with the name **ZhonePrivate**. This community has admin access to the system. Zhone recommends that you configure community names and access profiles to prevent unauthorized access to the system.

The **community-profile** specifies the community name and an access level for SNMP manager to access the system. It can also optionally specify a **community-access-profile** which is used to verify the source IP address of the SNMP manager. The system supports up to 50 different access lists.

The following community access levels are supported:

- **noaccess**—the community has no access.
- **read**—the community has read-only access to the system, with the exception of information in the **community-profile** and **community-access-profile**.
- **readandwrite**—the community has read/write access to the system, with the exception of information in the **community-profile** and **community-access-profile**.
- **admin**—the community has read and write access to the entire system, including information in the **community-profile** and **community-access-profile**. Note that the ZMS requires admin access to manage the system.

Create a community profile



Note: Configuring a community profile disables the **ZhonePrivate** default community name. If you do change the community name, you must change the name in ZMS or the device will become unmanageable.

The following example defines a community name **public** with read-only privileges:

```
zSH> new community-profile 1
Please provide the following: [q]uit.
community-name: -----> {}: public
```



```
permissions: -----> {read}:
access-table-index: -> {0}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Create community access profiles

The following example defines a community name **private** with read/write privileges and also creates an access list to verify that the SNMP client attempting to access the MXK is coming from known IP addresses 192.168.9.10 and 192.168.11.12:

First, create an access list for the first IP address:

```
zSH> new community-access-profile 2
Please provide the following: [q]uit.
access-table-index: -> {0}: 1
ip-address: -----> {0.0.0.0}: 192.168.9.10
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Then, create an access list for the second IP address with the same **access-table-index** (1):

```
zSH> new community-access-profile 3
Please provide the following: [q]uit.
access-table-index: -> {0}: 1
ip-address: -----> {0.0.0.0}: 192.168.11.12
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Finally, create a **community-profile** that specifies the community name, and uses the same **access-table-index** (1) as defined in the two **community-access-profiles** you just created:

```
zSH> new community-profile 4
Please provide the following: [q]uit.
community-name: -----> {}: private ZMS must include this name
permissions: -----> {read}: readandwrite
access-table-index: -> {0}: 1
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Configure traps

The **trap-destination** profile defines a trap recipient the MXK will send traps to. To configure a trap destination you need:

- the IP address of the SNMP trap server

- the community name the trap recipient expects

The other parameters in the **trap-destination** profile can be left at their default values. The following example configures a trap recipient with the IP address 192.168.3.21:

```
zSH> new trap-destination 32
Please provide the following: [q]uit.
trapdestination: -> {0.0.0.0}: 192.168.3.21
communityname: ---> {}: public
resendseqno: -----> {0}:
ackedseqno: -----> {0}:
traplevel: -----> {low}:
traptype: -----> {(null)}: 0
trapadminstatus: -> {enabled}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```



Note: When ZMS configures a device, a trap destination profile is automatically created.

MXK port management

This section describes port management on the MXK:

- [Port command overview, page 111](#)
- [View the administrative and operational states of ports with the port status and port show command, page 112](#)
- [View DDM data on Ethernet SFPs with the port show command, page 112](#)
- [Change port administrative states with the port testing, up, down, or bounce commands, page 115](#)
- [Port descriptions on the MXK, page 117](#)
- [Port mirroring, page 123](#)
- [Ethernet Jumbo Frames, page 126](#)

Port command overview

The **port** command has various administrative functions and is used to:

- alter the administrative status of a physical port or virtual interface on the MXK with the **port up**, **port down**, **port bounce**, or **port testing** commands. See [Port descriptions on the MXK on page 117](#).
- verify the administrative status of a physical port or virtual interface on the MXK with the **port show** command. See [View the administrative and operational states of ports with the port status and port show command on page 112](#).
- View DDM data on Ethernet SFPs with the **port show** command. See [View DDM data on Ethernet SFPs with the port show command on page 112](#).
- view the operational status, speed, and duplex mode of Ethernet ports with the **port status** command. See [View the administrative and operational states of ports with the port status and port show command on page 112](#).
- associate a text string with a physical interface, including bond groups, with the **port description** set of commands. See [Port descriptions on the MXK on page 117](#).
- display or clear various statistical information on Ethernet ports with the **port stats** command. See [MX\(P\)-160/260 enhanced Ethernet port statistics on page 369](#).
- set the severity level of alarms on Ethernet ports with the **port config alarm** command. See [Settable alarm severity for Ethernet ports on page 1165](#).

- configure jumbo Ethernet frames with the port config command and verify the change with the port show command. See [Ethernet Jumbo Frames on page 126](#)

View the administrative and operational states of ports with the port status and port show command

port status and port show command

Use the **port status** command to view the operational status, speed, and duplex mode of an Ethernet port.



Note: The **port status** command is only valid for Ethernet ports.

```
zSH> port status 1-6-1-0/eth
Operational status : Up
Rate in Mbps      : 1000
Duplex            : Full
```

Use the **port show** command to view the administrative status of a port or interface.

```
zSH> port show 1-2-1-0/vdsl
Interface 1-2-1-0/vdsl
Physical location: 1/2/1/0/vdsl
Administrative status: up
```

```
zSH> port show 1-a-2-0/eth
Interface 1-a-2-0/eth
Physical location: 1/a/2/0/eth
Administrative status: up
Port type specific information:
Link state mirroring not configured.
```

```
zSH> port show 1-6-1-0-eth/bridge
Interface 1-6-1-0-eth/bridge
Administrative status: up
```

View DDM data on Ethernet SFPs with the port show command

This section describes DDM on SFPs for Ethernet:

- [DDM data on Ethernet SFPs overview, page 113](#)
- [DDM data on Ethernet line card Ethernet SFPs, page 113](#)
- [DDM data on uplink card Ethernet SFPs, page 114](#)

DDM data on Ethernet SFPs overview

Digital Diagnostic Monitoring (DDM) provides SFP diagnostic data. For SFPs that support DDM, the SFP transceiver measures the temperature, supply voltage, transmit bias current, transmit power, and the receive power on the SFP.

Use the **port show interface/type** to display DDM data on Ethernet ports using SFPs that support DDM. [Table 8](#) describes the DDM data fields displayed.

For information on GPON DDM, see [Received Signal Strength Indication \(RSSI\) and Digital Diagnostic Monitoring \(DDM\)](#), page 948.

Table 8: port show command output fields for DDM data on Ethernet ports

Field	Description
Temperature	Internally measured Transceiver Temperature in degrees celsius.
Voltage	Internally measured Transceiver Supply Voltage in hundredths of volts.
Tx Bias Current	Measured Tx Bias current in milliamperes.
Transmit Power	Measured Tx Output power in tenths of dB.
Receive Power	Measured Rx power in tenths of dB.

DDM data on Ethernet line card Ethernet SFPs

SFP supports DDM data on Ethernet line card.

```
zSH> port show 1-1-1-0/eth
Interface 1-1-1-0/eth
  Physical location:      1/1/1/0/eth
  Administrative status: up
  Port type specific information:
    Frame size: 0 bytes
    Ingress rate: 0 Kbps burst size: 0 Kbits
    Egress rate: 0 Kbps burst size: 0 Kbits
  DDM data:
    Temperature:         31c
    Voltage:              3.29v
    Tx bias current:      29mA
    Transmit power:       -2.3dBm
    Receive power:        0.2dBm
```

SFP does not support DDM data on Ethernet line card.

```
zSH> port show 1-2-1-0/eth
Interface 1-2-1-0/eth
  Physical location:      1/2/1/0/eth
  Administrative status: up
```

```
Port type specific information:
    Frame size: 0 bytes
    Ingress rate: 0 Kbps burst size: 0 Kbits
    Egress rate: 0 Kbps burst size: 0 Kbits
DDM not supported
```

SFP not present on the Ethernet port of the Ethernet line card.

```
zSH> port show 1-1-10-0/eth
Interface 1-1-10-0/eth
    Physical location:    1/1/10/0/eth
    Administrative status: down
    Port type specific information:
        Frame size: 0 bytes
        Ingress rate: 0 Kbps burst size: 0 Kbits
        Egress rate: 0 Kbps burst size: 0 Kbits
    SFP not present
```

DDM data on uplink card Ethernet SFPs

Ethernet port on uplink card with SFP that supports DDM data.

```
zSH> port show 1-a-2-0/eth
Interface 1-a-2-0/eth
    Physical location:    1/a/2/0/eth
    Administrative status: up
    Port type specific information:
        Frame size: 0 bytes
        Ingress rate: 0 Kbps burst size: 0 Kbits
        Egress rate: 0 Kbps burst size: 0 Kbits
    DDM data:
        Temperature:      24c
        Voltage:           3.31v
        Tx bias current:   27mA
        Transmit power:    -2.1dBm
        Receive power:     0.1dBm
```

Ethernet port on uplink card with without SFP.

```
zSH> port show 1-a-3-0/eth
Interface 1-a-3-0/eth
    Physical location:    1/a/3/0/eth
    Administrative status: up
    Port type specific information:
        Frame size: 0 bytes
        Ingress rate: 0 Kbps burst size: 0 Kbits
        Egress rate: 0 Kbps burst size: 0 Kbits
    SFP not present
```

Ethernet port on uplink card with SFP that does not support DDM data.

```
zSH> port show 1-a-5-0/eth
Interface 1-a-5-0/eth
    Physical location:    1/a/5/0/eth
```

```

Administrative status: up
Port type specific information:
    Frame size: 0 bytes
    Ingress rate: 0 Kbps burst size: 0 Kbits
    Egress rate: 0 Kbps burst size: 0 Kbits
DDM not supported

```

Ethernet craft port on uplink card that does not use SFPs.

```

MXK-23> port show 1-a-1-0/eth
Interface 1-a-1-0/eth
    Physical location:    1/a/1/0/eth
    Administrative status: up
    Port type specific information:
        Frame size: 0 bytes
        Ingress rate: 0 Kbps burst size: 0 Kbits
        Egress rate: 0 Kbps burst size: 0 Kbits
    No DDM data available from ethernet port

```

Change port administrative states with the port testing, up, down, or bounce commands

port testing command

Use the **port testing** command to set the administrative state to testing on an Ethernet port.

```

zSH> port testing 1-6-1-0/eth
1-6-1-0/eth set to admin state TESTING

```

Verify the state.

```

zSH> port show 1-6-1-0/eth
Interface 1-6-1-0/eth
    Physical location:    1/6/1/0/eth
    Description:         Test
    Administrative status: testing
    Port type specific information:
        Link state mirroring not configured.

```

Use the **port testing** command to set the administrative state to testing on an VDSL2 port.

```

zSH> port testing 1-1-1-0/vdsl
1-1-1-0/vdsl set to admin state TESTING

```

Verify the state.

```

zSH> port show 1-1-1-0/vdsl
Interface 1-1-1-0/vdsl
    Physical location:    1/1/1/0/vdsl
    Administrative status: testing

```

port up command

Use the **port up** command to set the administrative state to up on an Ethernet port.

```
zSH> port up 1-6-1-0/eth
1-6-1-0/eth set to admin state UP
```

Verify the state.

```
zSH> port show 1-6-1-0/eth
Interface 1-6-1-0/eth
  Physical location:    1/6/1/0/eth
  Description:         Test
  Administrative status: up
  Port type specific information:
    Link state mirroring not configured.
```

Use the **port up** command to set the administrative state to up on an VDSL2 port.

```
zSH> port up 1-1-1-0/vdsl
1-1-1-0/vdsl set to admin state UP
```

Verify the state.

```
zSH> port show 1-1-1-0/vdsl
Interface 1-1-1-0/vdsl
  Physical location:    1/1/1/0/vdsl
  Administrative status: up
```

port down command

Use the **port down** command to set the administrative state to up on an Ethernet port.

```
zSH> port down 1-a-2-0/eth
1-a-2-0/eth set to admin state DOWN
```

Verify the state.

```
zSH> port show 1-a-2-0/eth
Interface 1-a-2-0/eth
  Physical location:    1/a/2/0/eth
  Administrative status: down
  Port type specific information:
    Link state mirroring not configured.
```

Use the **port down** command to set the administrative state to up on an VDSL2 port.

```
zSH> port down 1-1-1-0/vdsl
1-1-1-0/vdsl set to admin state DOWN
```

Verify the state.


```
zSH> port show 1-1-1-0/vdsl
Interface 1-1-1-0/vdsl
    Physical location:    1/1/1/0/vdsl
    Administrative status: down
```

port bounce command

Use the **port bounce** command to perform a down operation followed by an up operation on an Ethernet port.

```
zSH> port bounce 1-a-2-0/eth
1-a-2-0/eth set to admin state DOWN
1-a-2-0/eth set to admin state UP
```

Verify the state.

```
zSH> port show 1-a-2-0/eth
Interface 1-a-2-0/eth
    Physical location:    1/a/2/0/eth
    Administrative status: up
    Port type specific information:
        Link state mirroring not configured.
```

Use the **port bounce** command to perform a down operation followed by an up operation on a VDSL2 port.

```
zSH> port bounce 1-1-1-0/vdsl
1-1-1-0/vdsl set to admin state DOWN
1-1-1-0/vdsl set to admin state UP
```

Verify the state.

```
zSH> port show 1-1-1-0/vdsl
Interface 1-1-1-0/vdsl
    Physical location:    1/1/1/0/vdsl
    Administrative status: up
```

Port descriptions on the MXK

This section describes port descriptions:

- [Port description rules, page 117](#)
- [Add, modify, list, and delete a port description, page 118](#)
- [Search port descriptions, page 122](#)

Port description rules

The MXK has a port description field, which provides a mapping between the physical port, or bonded interface, or bridge and a subscriber. This mapping improves MXK management without requiring extra documents to provide the mapping. Port description information can be entered for ports, bridges, or bond groups. Port description information is also searchable.

The rules for entering a port description are:

- Port descriptions do not have to be unique.
- The port description field is a text field 64 characters long.
- Any characters can be used including spaces, \$, @, -, ., etc. The only characters not supported are the double quote, ‘ “ ’ which is a delimiter to identify the beginning and end of the text string, the carat ‘^’, and the question mark ‘?’.
- Port descriptions are associated with physical ports and not logical interfaces. For bonding technologies port descriptions are supported both on the physical port and the bond group, so if you want to use a keyword such as a company name to group interfaces.
- Even though port descriptions are searchable, you cannot perform commands using port description. For example, you can not use a command like “bridge modify circuitName ...”

Add, modify, list, and delete a port description

The **port description add** command associates a text string with a physical interface (which includes bond groups):

```
port description add <physical interface> <text string>
```



Note: Port descriptions do not need to be unique. If one customer has many lines, they may all have the same port description. You may also use the port description field as a means to group interfaces. See [Search port descriptions, page 122](#).

Add a port description to a port

To add a port description with spaces to a port, enter:

```
zSH> port description add 1-6-1-0/eth "510 555 5555"
```

In this case, the port description has spaces so quotes are needed.

To verify the port description, enter:

```
zSH> port show 1-6-1-0/eth
Interface 1-6-1-0/eth
  Physical location:    1/6/1/0/eth
  Description:         510 555 5555
  Administrative status: up
  Port type specific information:
    Link state mirroring not configured.
```

To add a port description without spaces to a port, enter:

```
zSH> port description add 1-6-2-0/eth BusinessPark
```

To verify the port description enter:

```
zSH> port show 1-6-2-0/eth
Interface 1-6-2-0/eth
  Physical location:    1/6/2/0/eth
  Description:         BusinessPark
  Administrative status: up
  Port type specific information:
    Link state mirroring not configured.
```

Add a port description to a GPON OLT port and ONU port

Both the GPON OLTs and the ONUs can have port descriptions.

To add a port description on a GPON OLT, enter:

```
zSH> port description add 1-4-1-0/gponolt SFO
```

To verify the port description, enter:

```
zSH> port show 1-4-1-0/gponolt
Interface 1-4-1-0/gponolt
  Physical location:    1/4/1/0/gponolt
  Description:         SFO
  Administrative status: up
```

To add a port description to a GPON ONU, enter:

```
zSH> port description add 1-4-1-1/gpononu "business 1 555-555-5555"
```

In this case, a port description is added to ONU 1 on OLT 1.

To verify the port description, enter:

```
zSH> port show 1-4-1-1/gpononu
Interface 1-4-1-1/gpononu
  Description:         business 1 555-555-5555
  Administrative status: up
```

Add a port description to a bridge

The port description must be added to the physical port of a bridge configuration. A port description can be added to the physical port of an existing bridge configuration or the port description can be added to the physical port that is then configured as a bridge.

View existing bridges:

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge
St  Table Data
-----
dwn                200      1/6/2/0/eth      1-6-2-0-eth/bridge
DWN
```

```

upl          Tagged 200    1/a/8/0/eth          ethernet8-200/bridge
DWN S VLAN 200 default
2 Bridge Interfaces displayed

```

Add the port description to the physical port of an existing bridge configuration, in this case the downlink bridge on Ethernet port 2:

```
zSH> port description add 1-6-2-0/eth "US Insurance Consortium, Inc."
```

Verify the port description on the downlink bridge:

```

zSH> bridge showdetail 1-6-2-0-eth/bridge
Bridge interface: 1-6-2-0-eth
    Administrative status: up          Operational status: down
    Blocked status: unblocked
    Type:dwn                          200
    Data:
Physical interface: 1-6-2-0/eth
    Administrative status: up          Operational status: down
Description: US Insurance Consortium, Inc.
Interface On Demand Stats State: enabled
Total Packet Statistics
    Received
        Unicast: 0
        Multicast: 0
        Broadcast: 0
        Bytes: --
    Sent
        Unicast: 0
        Multicast: 0
        Broadcast: 0
        Bytes: --
        Errors: 0
    Packet Storm Blocked
        Unicast: 0
        Multicast: 0
        Broadcast: 0
        Alarms: 0
Delta Packet Statistics - Collecting a 1 second data interval
    Received                               Sent
    Unicast  Multicast  Broadcast  Unicast  Multicast  Broadcast  Error
Delta      0           0           0         0         0           0         0
Rate      0           0           0         0         0           0         0
    IGMP Received                               IGMP Transmitted
    GenQuery SpecQuery vxReport  Leave  GenQuery SpecQuery vxReport  Leave
    0/0      0/0      0/0      0       0/0      0/0      0/0      0
IGMP misc: unknown= 0 errorRx= 0 actChans= 0 actHosts= 0

```

Add a port description to a bond group

View the existing bond groups:

```

zSH> bond show all
Bond Groups

```

Slot	GrpId	Type	State	Name	Desc
5	124	efmbond	OOS	1-5-124-0	-
5	25	efmbond	OOS	bond-0025	-

To add a port description to an existing bond group enter:

```
zSH> port description add bond-0025/efmbond "Mary's Nail Shop"
```

To verify the port description on the bond group enter:

```
zSH> bond show group bond-0025/efmbond
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
5	25	efmbond	OOS	bond-0025	Mary's Nail Shop
Group Members					
Slot	Port	Type	State	Name	Desc
5	2	shdsl	OOS	1-5-2-0	-
5	4	shdsl	OOS	1-5-4-0	-
5	3	shdsl	OOS	1-5-3-0	-

Or enter:

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
5	124	efmbond	OOS	1-5-124-0	-
5	25	efmbond	OOS	bond-0025	Mary's Nail Shop

Modify a port description

The port description modify command allows you to edit an existing port description.

```
port description modify <physical interface> <text string>
```

Enter a port description:

```
zSH> port description add 1-4-1-2/gpononu "Cafe Roma"
```

Verify the description:

```
zSH> port show 1-4-1-2/gpononu
```

```
Interface 1-4-1-2/gpononu
  Description:      Cafe Roma
  Administrative status: up
```

Modify the description on the same port:

```
zSH> port description modify 1-4-1-2/gpononu "Cafe Barrone"
```

Verify the change:

```
zSH> port show 1-4-1-2/gpononu
```

```
Interface 1-4-1-2/gpononu
  Description:      Cafe Barrone
```

Administrative status: up

Port description list

The **port description list** command will list the descriptions on a particular port.

```
zSH> port description list 1/4/1
```

Interface	Description
1-4-1-0/gponolt	SFO
1-4-1-1/gpononu	business 1 555-555-5555
1-4-1-2/gpononu	Cafe Barrone

Port description delete

The port description delete command removes the port description from the physical interface.

```
port description delete <physical interface>
```

To view the port description on a physical port enter:

```
zSH> port show 1-6-2-0/eth
```

```
Interface 1-6-2-0/eth
  Physical location:    1/6/2/0/eth
  Description:         US Insurance Consortium, Inc.
  Administrative status: up
  Port type specific information:
    Frame size: 0 bytes
    Ingress rate: 0 Kbps burst size: 0 Kbits
    Engress rate: 0 Kbps burst size: 0 Kbits
  SFP not present
```

To delete the port description enter:

```
zSH> port description delete 1-6-2-0/eth
```

To verify the deletion enter:

```
zSH> port show 1-6-2-0/eth
```

```
Interface 1-6-2-0/eth
  Physical location:    1/6/2/0/eth
  Administrative status: up
  Port type specific information:
    Frame size: 0 bytes
    Ingress rate: 0 Kbps burst size: 0 Kbits
    Engress rate: 0 Kbps burst size: 0 Kbits
  SFP not present
```

Search port descriptions

The **port description find** command provides a textual search which allows you search for a text string within the port description fields. The display

show the description and the physical location. If multiple port descriptions have the same text string they will all be displayed

port description find <text string>

```
zSH> port description find 510
```

Results for 510

Description: 510 555 5555

Interface: 1-13-1-0/eth

```
zSH> port description find "business 1 555-555-5555"
```

Results for business 1 555-555-5555

Description: business 1 555-555-5555

Interface: 1-4-1-1/gpononu



Note: Notice that for search items which do not have spaces the quotation marks are unnecessary.

Port mirroring

The MXK provides port mirroring as a diagnostic tool used to troubleshoot packet movement on uplink ports.

The rules for port mirroring are:

- The MXK supports one mirror at a time.
- All mirrored ports must be on the same uplink card even in a redundant configuration.
- Any Ethernet port can be mirrored to any other Ethernet port on the same card except for the management 10/100 Ethernet port.
- When a port is a member of a link aggregation group, either the link aggregation group or one port in the link aggregation group can be mirrored.



Note: If more than one port needs to be mirrored, you must put the ports in a link aggregation group. The ports must stay in the link aggregation group for mirroring to continue.

port mirror command syntax

The syntax for the **port mirror** command is:

```
port mirror <from-interface> <to-interface> <vlan
<vlanId>> <in|out|both|off>
```

Table 9: Variables for the port mirror command

Variable	Definition
from-interface	The interface to mirror.
to-interface	Where to send the packets.
vlanID	The outer VLAN tag.
in	Mirror the incoming packets.
out	Mirror the outgoing packets.
both	Mirror both the incoming and outgoing packets.
off	Disable port mirroring for the port interface.

Create a mirrored port on the uplink card

Case 1: Configuring an uplink Ethernet port to mirror packets entering a 100/1000 Ethernet port to a 100/1000 Ethernet port

- 1 In this case, both ports are 100/1000 Ethernet ports.

```
zSH> port mirror 1-a-9-0/eth 1-a-11-0/eth vlan 200 in
```

This example enables port mirroring to send packets entering *1-a-9-0/eth* to *1-a-11-0/eth* on VLAN 200.

- 2 When necessary, turn port mirroring *off*.

```
zSH> port mirror 1-a-9-0/eth 1-a-11-0/eth vlan 200 off
```

Case 2: Configuring an uplink Ethernet port to mirror packets leaving a 10G Ethernet port to a 100/1000 Ethernet port

- 1 In this case, port *1-a-2-0/eth* is a 10G Ethernet port, and port *1-a-9-0/eth* is a 100/1000 Ethernet port.

```
zSH> port mirror 1-a-2-0/eth 1-a-9-0/eth vlan 700 out
```

This example enables port mirroring to send packets leaving *1-a-2-0/eth* to *1-a-9-0/eth* on VLAN 700

- 2 When necessary, turn port mirroring *off*.

```
zSH> port mirror 1-a-2-0/eth 1-a-9-0/eth vlan 700 off
```


Case 3: Configuring an uplink Ethernet port in a link aggregation group to mirror packets entering and leaving the ports in a linkagg group to a 100/1000 GE Ethernet port

- 1 Verify the ports in the link aggregation group.

```
zSH> linkagg show
```

```
LinkAggregations:
```

slot	unit	ifName	partner: Sys			Pri	grp ID	status	agg mode
a*	1	1-a-1-0	00:00:00:00:00:00			0x0	0x0	OOS	Active
		links	slot	port	subport		status		
		1-a-7-0	a	7	0		ACT		
		1-a-6-0	a	6	0		ACT		
b	1	1-b-1-0	00:00:00:00:00:00			0x0	0x0	OOS	Active
		links	slot	port	subport		status		
		1-b-7-0	a	7	0		DSA		
		1-b-6-0	b	6	0		DSA		

```
global linkagg group red type: red
```

- 2 In this case, *1-a-1-0/linkagg* is the linkagg group and *1-a-8-0/eth* is the 100/1000 GE Ethernet port.

```
zSH> port mirror 1-a-1-0/linkagg 1-a-8-0/eth vlan 900 both
```

This example enables port mirroring to send packets both entering and leaving port *1-a-7-0/eth* and port *1-a-6-0/eth* in the link aggregation group to port *1-a-8-0/eth* on VLAN 900.

- 3 When necessary, turn port mirroring *off*.

```
zSH> port mirror 1-a-1-0/linkagg 1-a-8-0/eth vlan 900 off
```

Case 4: Configuring an uplink Ethernet port to mirror packets entering and leaving a 100/1000 GE Ethernet port to a 10G Ethernet port

- 1 In this case, port *1-a-11-0/eth* is a 100/1000 GE Ethernet port and *1-a-2-0/eth* is a 10G Ethernet port.

```
zSH> port mirror 1-a-11-0/eth 1-a-2-0/eth vlan 800 both
```

This example enables port mirroring to send packets both entering and leaving *1-a-11-0/eth* to *1-a-2-0/eth*.

- 2 When necessary, turn port mirroring *off*.

```
zSH> port mirror 1-a-11-0/eth 1-a-2-0/eth vlan 800 off
```

Ethernet Jumbo Frames

Jumbo Ethernet frames are defined as frames that exceed 1500 bytes of payload. Jumbo Ethernet frames are usually up to 9000 bytes of payload and are frequently used by data centers to provide lower overhead Ethernet connectivity. Enterprise Ethernet, carrier Ethernet, and access networks are now frequently requiring jumbo Ethernet frames.

```
zSH> port show 1-1-1-0/eth
Interface 1-1-1-0/eth
    Physical location:      1/1/1/0/eth
    Administrative status: up
    Port type specific information:
        Frame size: 0 bytes
        Ingress rate: 0 Kbps burst size: 0 Kbits
        Egress rate: 0 Kbps burst size: 0 Kbits
    DDM not supported

zSH> port config 1-1-1-0/eth maxframe 9120
Setting max frame size to: 9120 bytes.
Interface 1-1-1-0/eth configured for max frame size of 9120.
```

```
zSH> port show 1-1-1-0/eth
Interface 1-1-1-0/eth
    Physical location:      1/1/1/0/eth
    Administrative status: up
    Port type specific information:
        Frame size: 9120 bytes
        Ingress rate: 0 Kbps burst size: 0 Kbits
        Egress rate: 0 Kbps burst size: 0 Kbits
    DDM not supported
```

```
zSH> get ether 1/1/1/0
ether 1/1/1/0
autonegstatus: ----> {enabled}
mauType: -----> {mau1000basetfd}
restart: -----> {norestart}
ifType: -----> {mau1000basetfd}
autonegcap: -----> {b100baseTX+b100baseTXFD+b1000baseT+b1000baseTFD}
remotefault: -----> {noerror}
clksrc: -----> {automatic}
pauseFlowControl: -> {disabled}
aggregationMode: --> {on}
linkStateMirror: --> {0/0/0/0/0}
maxFrameSize: -----> {9120}
ingressRate: -----> {0}
```

```

ingressBurstSize: -> {0}
egressRate: -----> {0}
egressBurstSize: --> {0}

```

zSH> **port config 1-1-1-0/eth maxframe 0**

Interface 1-1-1-0/eth configured to default frame size of 2048 bytes.

zSH> **get ether 1/1/1/0**

```

ether 1/1/1/0
autonegstatus: ----> {enabled}
mauType: -----> {mau1000basetfd}
restart: -----> {norestart}
ifType: -----> {mau1000basetfd}
autonegcap: -----> {b100baseTX+b100baseTXFD+b1000baseT+b1000baseTFD}
remotefault: -----> {noerror}
clksrc: -----> {automatic}
pauseFlowControl: -> {disabled}
aggregationMode: --> {on}
linkStateMirror: --> {0/0/0/0/0}
maxFrameSize: -----> {0}
ingressRate: -----> {0}
ingressBurstSize: -> {0}
egressRate: -----> {0}
egressBurstSize: --> {0}

```

MXK security

This section describes the MXK's security features including Radius support, Secure Shell (SSH), Secure File Transfer Protocol (SFTP), HTTPS and port access security.

- [MXK security \(SSH, SFTP, and HTTPS\), page 128](#)
- [Port access security, page 132](#)
- [Radius support, page 134](#)



Note: For security reasons, host keys are not accessible via SNMP and cannot be saved/restored with the dump command.

MXK security (SSH, SFTP, and HTTPS)

This section covers the security on the MXK:

- [Enable security on the MXK, page 128](#)
- [DSA and RSA keys, page 130](#)
- [Tested MXK SSH clients, page 130](#)
- [Encryption-key commands, page 131](#)

Enable security on the MXK

The **system 0** profile provides a **secure** parameter which allows only secure communication for management activities. When security is enabled, the MXK uses the following protocols:

- Secure File Transfer Protocol (SFTP)
- Secure shell (SSH)
- HTTPS (HTTP secure)

[Table 10](#) describes which protocols are allowed when the **secure** parameter is *enabled* and which protocols are allowed when the **secure** parameter is *disabled*.

Table 10: Protocols for the secure parameter

Disabled	Enabled
TFTP, FTP	SFTP
Telnet, SSH	SSH
HTTP	HTTPS

Enabling security on the MXK

To enable the security parameter enter **update system 0** on the MXK, change the secure parameter from *disabled* to *enabled*, then save the file:



Note: After enabling the **secure** parameter, HTTPS and changes to the Web UI take affect after the next reboot. SSH and SFTP do not require a reboot.

```
zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {false}:
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: ---> {false}:
configsyncpriority: ---> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: ---> {}:
configsyncstatus: -----> {syncinitializing}:
configsyncuser: -----> {}:
configsyncpasswd: -----> ** private **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {0.0.0.0}:
alternateipaddress: ---> {0.0.0.0}:
countryregion: -----> {us}:
primaryclocksource: ---> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}: enabled
webinterface: -----> {enabled}:
options: -----> {NONE(0)}:
reservedVlanIdStart: --> {0}:
reservedVlanIdCount: --> {0}:
snmpVersion: -----> {snmpv2}:
persistentLogging: ---> {disabled}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

DSA and RSA keys

The MXK automatically creates a Digital Signature Algorithm (DSA), a standard for digital signatures, and supports RSA, an algorithm for public-key cryptography. The DSA and RSA host keys for the server are persistently stored in the encryption-key profile. In order to manage the host keys, use the CLI command **encryption-key**.

RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key.

When the system first boots, it will try to load the existing DSA and RSA keys. If they do not exist, the system creates a 512 bit DSA key.

The CLI **encryption-key** command can be used to view current keys, create a new key, regenerate keys that may have been compromised, and delete keys.

To create a new key enter:

```
zSH> encryption-key add rsa 1024
Generating key, please wait ... done.
```



Note: Generating keys is computationally intensive. The longer the key, the longer it takes to generate. Wait until the system shows that key generation is completed before you continue.

To view the new key just created enter:



Note: The **encryption-key show** command displays the keys that were generated and are available for use. The command does not show the actual keys.

```
zSH> encryption-key show
Index Type      Length
-----
 1   dsa         512
 2   rsa        1024
```

To regenerate a key that might have been compromised enter:

```
zSH> encryption-key renew dsa
Generating key, please wait ... done.
```

To delete an encryption key enter:

```
zSH> encryption-key delete dsa
```

Tested MXK SSH clients

Secure Shell (SSH) is a command interface and protocol for securely getting access to a remote computer. SSH commands are encrypted and secure in two ways. Both ends of the client/server connection are authenticated using a

digital certificate, and passwords are protected by being encrypted. You can now connect to a MXK using the SSH client of your choice to encrypt the session. The MXK SSH2 only with the following SSH clients:

- OpenSSH
 - cygwin
 - Linux
 - Solaris
- Putty
- Teraterm
- SecureCRT
- Absolute Telnet

Encryption-key commands

encryption-key add

Adds an encryption key to the encryption-key profile.

Syntax `encryption-key add [rsa|dsa] [512|768|1024|2048]`

Options `rsa|dsa`
 Name and type of the encryption key.

`512|768|1024|2048`
 The number of bytes the key is set to.

encryption-key delete

Deletes an encryption key from the encryption-key profile.

Syntax `encryption-key delete [rsa|dsa]`

Options `rsa|dsa`
 Name and type of the encryption key.

encryption-key renew

Regenerates a compromised encryption key.

Syntax `encryption-key renew [rsa|dsa]`

Options `rsa|dsa`
 Name and type of the encryption key.

encryption-key show

Displays the current encryption keys.

Syntax `encryption-key show`

Port access security

The MXK provides security capabilities on the UDP/TCP ports which the MXK uses for management. Use the **port-access** profile to define the UDP/TCP port and the IP address or IP address subnet that allows access to that port.

The port access security feature is a white list mechanism. If a host's IP address is not specified in a **port-access** profile, users from that host cannot access on that port.

The management ports are:

- Telnet, port 23
- SSH, port 22
- HTTP, port 80
- HTTPS, port 443
- SNMP, port 161

In order to restrict access to the SNMP port, there must be a rule to allow the MXK its own SNMP access. See [Creating a port-access entry for the MXK to maintain SNMP access on page 134](#).

By default, **port-access** profiles do not exist and all ports are open. After a **port-access** profile is configured for a port all other IP addresses or subnets are blocked. This restriction only takes effect after the first **port-access** profile is created.



Note: Port access security is not independent from enabling secure mode for SFTP and SSH in **system 0**. If secure is enabled to provide SSH and SFTP while limiting Telnet access, and you have provided access with the **port-access** profile for Telnet to a device (or range of devices), the device(s) will not have access.

Up to 100 **port-access** profile entries can be created on a SLMS device.

Creating port-access profile entries

Create a **port-access** profile entry.

- 1** Create a new port-access entry by entering **new port-access n**, where n is an available entry ID number.
- 2** In the **portNumber** parameter enter the port number.
- 3** In the **srcAddr** parameter enter the IP address or first IP address of the subnet.
- 4** In the **netMask** parameter enter 255.255.255.255 for a single IP address mask, or a subnet mask for a subnet.

Creating a port-access entry for a specific IP address

Create a new **port-access** profile and specify the port number, host/network IP address to be granted access, and the one address netmask (255.255.255.255, which really means an exact mask of the IP address given) applied to the IP address to allow access to a single IP address.



Note: To create port access protection for both HTTP and HTTPS, create port access entries for port 80 and port 443.

```
zSH> new port-access 1
Please provide the following: [q]uit.
portNumber: -> {0}: 80
srcAddr: ---> {0.0.0.0}: 172.16.42.1
netMask: ---> {0.0.0.0}: 255.255.255.255
.....S=
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

This example creates **port-access** entry 1 on HTTP port 80 and allows hosts on the 172.16.42.1 network to have HTTP access to the MXK.

Creating a port-access entry for a subnet

Create a new **port-access** profile and specify the Telnet port number, initial host/network IP address to be granted access, and the netmask applied to the IP address to allow access to a range of IP addresses.



Note: Typically, only port 23 is used for Telnet access.

```
zSH> new port-access 2
Please provide the following: [q]uit.
portNumber: -> {0}: 23
srcAddr: ---> {0.0.0.0}: 172.16.41.0
netMask: ---> {0.0.0.0}: 255.255.255.0
.....S=
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

This example creates **port-access** entry 2 on Telnet port 23 and allows hosts on the 172.16.41.xx network to Telnet to the MXK.

Displaying port-access profile entries

Display configured **port-access** profile entries with the **list** command:

```
zSH> list port-access
port-access 1
1 entry found.
```

Modifying port-access profile entries

Modify a configured **port-access** profile entry with the **update** command.
This example changes the entry's source IP address to 172.16.40.0:

```
zSH> update port-access 2
portNumber: -> {23}
srcAddr: ---> {172.16.41.0} 172.16.40.0
netMask: ---> {255.255.255.0}
1 entry found.
.....
Save new record? [s]ave, [c]hange or [q]uit: s
Updated record saved.
```

Displaying port-access profile entries

To display configured **port-access** profile entries use the **list** command:

```
zSH> list port-access
port-access 1
1 entry found.
```

Creating a port-access entry for the MXK to maintain SNMP access

Create a new **port-access** profile and specify the SNMP port number (161) then 127.0.0.0 as the IP address for the subnet and a subnet mask of 255.0.0.0.

```
zSH> new port-access 10
Please provide the following: [q]uit.
portNumber: -> {0}: 161
srcAddr: ---> {0.0.0.0}: 127.0.0.0
netMask: ---> {0.0.0.0}: 255.0.0.0
.....S=
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Radius support

The MXK supports local and RADIUS (Remote Authentication Dial In User Service) access authentication. The MXK can be configured for local authentication, RADIUS authentication, or RADIUS then local authentication. RADIUS users are configured with the Service-Type attribute as Administrative-User or NAS-Prompt-User. RADIUS is used for only login authentication, not severity levels.

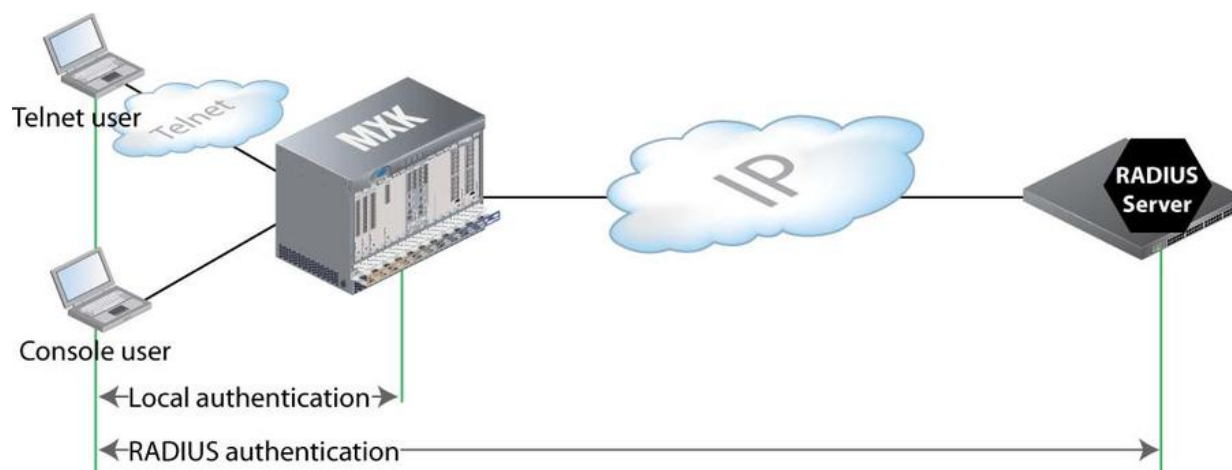
[Table 11](#) shows the mapping of service-type to MXK permissions.

Table 11: Service type mapping to MXK permissions

Service-Type Attribute	MXK permissions
Administrative-User	admin, zhonedebug, voice, data, manuf, database, systems, tools, useradmin
NAS-Prompt-User	admin, voice, data, manuf, database, systems, tools, useradmin

When establishing a connection to the MXK with RADIUS authentication, the MXK passes RADIUS information securely to the RADIUS server. The RADIUS server then authenticates the user and either allows or denies access to the MXK. If access is denied and the local authentication option is also configured, the MXK then authenticates access based on the locally configured users and passwords. For logins and failed logins, a console message is generated with user ID and IP address of the device from which the login originated. Failed logins also are logged as alert level messages in the MXK system log file.

By default, RADIUS access uses the UDP port 1812 for authentication. This parameter can be changed in the radius-client profile.

Figure 5: MXK RADIUS authentication

Note: Follow the RADIUS server guidelines for RADIUS configuration instructions. For example, when using the MXK with the FreeRadius server:

- Create only one entry in the *clients.conf* file for each subnet or individual MXK. For individual MXKs, the IP in this file must match the IP address of the outbound interface used by the MXK to connect to the RADIUS server.
- The MXK uses the value stored in the RADIUS *system.sysname* file for the NAS-Identifier attribute.

- The shared-secret in the MXK **radius-client** profile, must exactly match the shared-secret in the RADIUS client entry.

Configuring RADIUS support

The MXK can be configured for local authentication, RADIUS authentication, or RADIUS then local authentication. Multiple **radius-client** profiles can be defined using the index and subindex numbers. This index scheme can be used to create index numbers for groups of RADIUS servers. When an index number is specified in the system profile, the MXK attempts authentication from each RADIUS server in that group in sequential order of the subindex numbers.

To configure RADIUS support:



Note: Before beginning this procedure, ensure that the MXK has IP connectivity to the RADIUS server.

- 1 Update the RADIUS server with settings for the Zhone prompts.
- 2 Create a **radius-client** profile on the MXK with the desired index number and RADIUS settings for server name, shared secret, number of retries, and other parameters. The first number in the index is used to group **radius-client** profiles so multiple profiles can be assigned to a MXK. The second number in the index specifies the order in which **radius-client** profiles are referenced. This example specifies the **radius-client** 1/1 with server name *radius1* and a **shared-secret** of *secret*. A DNS resolver must be configured in the system to resolve the server name and IP address. If a DNS resolver is not available, specify the IP address of the server. The index 1/1 specifies that this profile is the first profile in group 1.

```
zSH> new radius-client 1/1
Please provide the following: [q]uit.
server-name: ----> {}: radius1.test.com [DNS resolver must be configured in the system.]
udp-port: -----> {1812}:
shared-secret: --> {** password **}: secret
retry-count: ----> {5}:
retry-interval: -> {1}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
Record created.
```

Another method to reference the RADIUS server is by specifying the IP address. This example specifies the **radius-client** 1/1 with server IP address 172.24.36.148 and a **shared-secret** of *secret*. The index 1/1 specifies that this profile is the first profile in group 1.

```
zSH> new radius-client 1/1
Please provide the following: [q]uit.
server-name: ----> {}: 172.24.36.248
udp-port: -----> {1812}:
shared-secret: --> {** password **}: secret
retry-count: ----> {5}:
```

```

retry-interval: -> {1}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
Record created.

```

- 3 Create another **radius-client** profile on the MXK with the desired RADIUS settings for server name, shared secret, number of retries, and other parameters. This example specifies the **radius-client 1/2** with server IP address 172.24.36.148 and a **shared-secret** of *secret*. The index 1/2 specifies that this profile is the second profile in group 1.

```

zSH> new radius-client 1/2
Please provide the following: [q]uit.
server-name: ----> {}: 172.24.36.249
udp-port: -----> {1812}:
shared-secret: --> {** password **}: secret
retry-count: ----> {5}:
retry-interval: -> {1}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
Record created.

```

Create additional **radius-client** profiles for each additional RADIUS server to be assigned to this MXK.

- 4 In the system profile on the MXK, set the desired user authentication method and specify the index of the radius profile to use. This example specifies the **radiusauthindex** of 1. This index is configured with two **radius-client** profiles (1/1, 1/2). The MXK first attempts authentication using the server specified in **radius-client 1/1**. If this authentication fails, the MXK attempts authentication using **radius-client 1/2** server. If this authentication also fails, the MXK then attempts authentication based on the authentication mode setting in the system profile. This example uses **radiusthenlocal**.



Caution: If the *radius* authentication mode is used, local authentication is disabled so the MXK may become inaccessible if IP connectivity to the RADIUS server is lost or other changes prevent the MXK from receiving RADIUS authentication.

```

zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {false}:
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:

```

```

configsyncoverflow: ---> {false}:
configsyncpriority: ---> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: ---> {}:
configsyncstatus: -----> {syncinitializing}:
configsyncuser: -----> {}:
configsyncpasswd: -----> ** private **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {0.0.0.0}:
alternateipaddress: ---> {0.0.0.0}:
countryregion: -----> {us}:
primaryclocksource: ---> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}:
userauthmode: -----> {local}: radiusthenlocal
radiusauthindex: -----> {0}: 1
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {NONE(0)}:
reservedVlanIdStart: --> {0}:
reservedVlanIdCount: --> {0}:
snmpVersion: -----> {snmpv2}:
persistentLogging: -----> {disabled}
outletTemperatureHighThreshold: -> {65}
outletTemperatureLowThreshold: --> {-12}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

After completing the RADIUS configuration, the MXK displays console messages for RADIUS login and logout activity.

For users logging in through RADIUS, the system prompt appears as the *username@systemname*. For example, the system prompt for a basic user on a MXK using the default Zhone MXK system name will appear as *basicuser@Zhone mxk*. The system name is configured using the **sysname** parameter in the **system 0** profile.

MXK alarms

This section describes the following:

- [Alarm manager, page 139](#)
- [Alarm suppression, page 140](#)
- [Configurable high and low chassis temperature alarms, page 142](#)

Alarm manager



Note: For GPON ONU alarms, refer to [GPON Alarms and Traps on page 987](#). The **alarm show** command does not display GPON ONU alarms.

The MXK central alarm manager includes the ability to view the active alarms on the system (using the **alarm show** command) and the ability to store active alarms on the device. ZMS can use the alarms stored on the device to recreate the state of the alarms if it becomes disconnected.

The alarm command uses the following syntax:

alarm show [**summary**]

For example, the following command displays the number of current active alarms, the total number of alarms, the number of cleared alarms, as well as each active alarm and its severity:

```
zSH> alarm show
*****      Central Alarm Manager      *****
      ActiveAlarmCurrentCount           :11
      AlarmTotalCount                   :36
      ClearAlarmTotalCount               :25
      OverflowAlarmTableCount            :0

ResourceId      AlarmType      AlarmSeverity
-----
1-a-2-0/eth     linkDown      critical
1-a-3-0/eth     linkDown      critical
1-a-6-0/eth     linkDown      critical
1-a-7-0/eth     linkDown      critical
1-a-8-0/eth     linkDown      critical
1-a-9-0/eth     linkDown      critical
1-a-10-0/eth    linkDown      critical
1-a-11-0/eth    linkDown      critical
1-2-2-1/other   linkDown      minor
system          power_supply_b_failure      warning
system          not_in_redundant_mode       major
```

The **summary** option displays the number of current active alarms, the total number of alarms, the number of system cleared alarms:

```
zSH> alarm show summary
*****      Central Alarm Manager      *****
```

```
ActiveAlarmCurrentCount      :84
AlarmTotalCount              :137
ClearAlarmTotalCount         :53
OverflowAlarmTableCount      :0
```

The **alarm clear** command clears a transient alarm the system was unable to clear.



Caution: Alarms cleared with the **alarm clear** command will not be redisplayed if condition reoccurs. The alarm will redisplay only if the condition reoccurs, goes away, and then reoccurs.

```
zSH> alarm clear
Num  ResourceId      AlarmType      AlarmSeverity
-----
  1  1-a-2-0/eth      linkDown      critical
  2  1-a-3-0/eth      linkDown      critical
  3  1-a-4-0/eth      linkDown      critical
....
34  1-5-3-0/gponolt   linkDown      critical
35  1-5-4-0/gponolt   linkDown      critical
36  1-5-5-0/gponolt   linkDown      critical
37  1-5-6-0/gponolt   linkDown      critical
38  1-5-7-0/gponolt   linkDown      critical
39  1-5-8-0/gponolt   linkDown      critical
40  1-4-1-0-gponolt/sn-1  gpon_unassigned_serial_number  warning
```

Caution: use this option with discretion.
Alarm will not be redisplayed if condition reoccurs. Alarm will redisplay only if condition reoccurs, goes away, and then reoccurs.
Enter alarm number from list, or 'q' to quit:

The alarm clear command only clears alarms one at a time by the alarm number displayed in the *Num* column.

Alarm suppression

The alarm suppression feature allows alarm/LED notification and output to be disabled based on alarm severity level for existing and future alarms. When an alarm level is disabled, all existing alarms of that type are cleared from the system. Future alarms of that type do not set LEDs or alarm relays and are not displayed in alarm output.

Alarm suppression is also supported in ZMS.

[Table 12](#) lists the alarm suppression options and the resulting behaviors. By default, alarms for all severity levels are enabled.

Table 12: Alarm suppression options

Alarm Levels Enabled Setting	Alarm Behavior
critical+major+minor+warning	Enables all alarm levels. The default setting.

Table 12: Alarm suppression options (Continued)

Alarm Levels Enabled Setting	Alarm Behavior
critical+major+minor	Disables all warning alarms.
critical+major	Disables all minor, and warning alarms.
critical+major+warning	Disables all minor alarms.
critical+minor+warning	Disables all major alarms.
critical+minor	Disables all major and warning alarms.
critical+warning	Disables all major and warning alarms.
critical	Disables all major, minor, and warning alarms.
major	Disables all critical, minor, and warning alarms.
major+minor+warning	Disables all critical alarms.
major+minor	Disables all critical and warning alarms.
major+warning	Disables all critical and minor alarms.
minor	Disables all critical, major, and warning alarms.
minor+warning	Disables all critical and major alarms.
(no levels)	Disables all alarm levels.

This example disables alarm/LED notification and output for all current and future alarms with the severity levels minor and warning.

```

zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {false}:
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: ---> {false}:
configsyncpriority: ---> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: ---> {}:
configsyncstatus: -----> {syncinitializing}:
configsyncuser: -----> {}:
configsyncpasswd: -----> ** private **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {0.0.0.0}:
alternateipaddress: ---> {0.0.0.0}:

```

```

countryregion: -----> {us}:
primaryclocksource: ---> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}: critical+major
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {NONE(0)}:
reservedVlanIdStart: --> {0}:
reservedVlanIdCount: --> {0}:
snmpVersion: -----> {snmpv2}:
persistentLogging: ----> {disabled}
outletTemperatureHighThreshold: -> {65}
outletTemperatureLowThreshold: --> {-12}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Configurable high and low chassis temperature alarms

High and low temperature threshold parameters were added to the **system** profile:

```

zSH> show system
...
outletTemperatureHighThreshold:-> {35 - 65}
outletTemperatureLowThreshold:--> {-40 - 0}

```

Parameter defaults are:

```

zSH> get system 0
...
outletTemperatureHighThreshold: -> {65}
outletTemperatureLowThreshold: --> {-12}

```

A minor alarm is raised when the outlet temperature is at the **outletTemperatureHighThreshold**. Major alarm is raised when the outlet temperature is **outletTemperatureHighThreshold+5**. Critical alarm is raised when the outlet temperature is **outletTemperatureHighThreshold+10**. For example, if the **outletTemperatureHighThreshold** is configured as 35, alarms will be in the order of 35, 40, 45 for Minor, Major, and Critical. If the **outletTemperatureHighThreshold** is configured as 65, alarms will be in the order of 65, 70, 75 for Minor, Major, and Critical.

When the **outletTemperatureLowThreshold** is set and the outlet sensor reaches the configured temperature, a Minor alarm is raised.

Configuring high and low chassis temperature alarms

- 1 Configure the **outletTemperatureHighThreshold** and the **outletTemperatureLowThreshold** parameter in the **system 0** profile.

```

zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {true}:
zmsconnectionstatus: -----> {inactive}:
zmsipaddress: -----> {10.51.1.241}:
configsyncexists: -----> {false}:
configsyncoverflow: -----> {false}:
configsyncpriority: -----> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: -----> {10.51.1.118_4_1405380127627}:
configsyncstatus: -----> {synccomplete}:
configsyncuser: -----> {zmsftp}:
configsyncpasswd: -----> {** private **}: ** read-only **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {10.51.1.118}:
alternateipaddress: -----> {0.0.0.0}:
countryregion: -----> {us}:
primaryclocksource: -----> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -----> {true}:
voicebandwidthcheck: -----> {false}:
alarm-levels-enabled: -----> {critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {NONE(0)}:
reservedVlanIdStart: -----> {0}:
reservedVlanIdCount: -----> {0}:
snmpVersion: -----> {snmpv2}:
persistentLogging: -----> {disabled}:
outletTemperatureHighThreshold: --> {65}: 50
outletTemperatureLowThreshold: --> {-12}: 0
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

- 2 Verify the changes.

```

zSH> get system 0
system 0

```

```

syscontact: -----> {}
sysname: -----> {}
syslocation: -----> {}
enableauthtraps: -----> {disabled}
setserialno: -----> {0}
zmsexists: -----> {true}
zmsconnectionstatus: -----> {inactive}
zmsipaddress: -----> {10.51.1.241}
configsyncexists: -----> {false}
configsyncoverflow: -----> {false}
configsyncpriority: -----> {high}
configsyncaction: -----> {noaction}
configsyncfilename: -----> {10.51.1.118_4_1405380127627}
configsyncstatus: -----> {synccomplete}
configsyncuser: -----> {zmsftp}
configsyncpasswd: -----> ** private **
numshelves: -----> {1}
shelvesarray: -----> {}
numcards: -----> {3}
ipaddress: -----> {10.51.1.118}
alternateipaddress: -----> {0.0.0.0}
countryregion: -----> {us}
primaryclocksource: -----> {0/0/0/0/0}
ringsource: -----> {internalringsourcelabel}
revertiveclocksource: -----> {true}
voicebandwidthcheck: -----> {false}
alarm-levels-enabled: -----> {critical+major+minor+warning}
userauthmode: -----> {local}
radiusauthindex: -----> {0}
secure: -----> {disabled}
webinterface: -----> {enabled}
options: -----> {NONE(0)}
reservedVlanIdStart: -----> {0}
reservedVlanIdCount: -----> {0}
snmpVersion: -----> {snmpv2}
persistentLogging: -----> {disabled}
outletTemperatureHighThreshold: -> {50}
outletTemperatureLowThreshold: --> {0}

```

- 3 View the alarms sent in the console window when thresholds are met or exceeded or use the **alarm show** command.

View the alarm when the outlet temperature reaches the configured temperature high threshold.

zSH> **log ses on**

Logging is already enabled for this session.

zSH> JUL 28 09:57:36: alert : 1/a/12 : shelfctrl: Warning: Temperature is above 50 degrees C (122 F) threshold.

JUL 28 09:57:36: alert : 1/a/12 : shelfctrl: Outlet temp=50 degrees C (122 F)

JUL 28 09:57:36: alert : 1/a/1025: alarm_mgr: 01: a:00 Minor Chassis Temperature above 50 degrees C (122 F) threshold

zSH> alarm show

```

*****      Central Alarm Manager      *****
      ActiveAlarmCurrentCount           :21
      AlarmTotalCount                   :100
      ClearAlarmTotalCount              :79
      OverflowAlarmTableCount           :0

ResourceId      AlarmType      AlarmSeverity
-----
1-a-3-0/eth     linkDown      critical
1-a-4-0/eth     linkDown      critical
1-a-5-0/eth     linkDown      critical
1-a-6-0/eth     linkDown      critical
1-a-7-0/eth     linkDown      critical
1-a-8-0/eth     linkDown      critical
1-a-9-0/eth     linkDown      critical
1-a-10-0/eth    linkDown      critical
1-a-11-0/eth    linkDown      critical
1-b-1-0/eth     linkDown      critical
1-b-3-0/eth     linkDown      critical
1-b-4-0/eth     linkDown      critical
1-b-5-0/eth     linkDown      critical
1-b-6-0/eth     linkDown      critical
1-b-7-0/eth     linkDown      critical
1-b-8-0/eth     linkDown      critical
1-b-9-0/eth     linkDown      critical
1-b-10-0/eth    linkDown      critical
1-b-11-0/eth    linkDown      critical
system          temp_over_limit  minor
slot 13         card_removed   warning

```

View the alarm when the outlet temperature exceeds the configured temperature high threshold by +5.

```

zSH> JUL 28 10:02:45: alert : 1/a/12 : shelfctrl: Warning: Temperature is above
55 degrees C (131 F) threshold.
JUL 28 10:02:45: alert : 1/a/12 : shelfctrl: Outlet temp=55 degrees C (131 F)
JUL 28 10:02:45: alert : 1/a/1025: alarm_mgr: 01: a:00 Minor Updating
Temperature alarm severity
JUL 28 10:02:45: alert : 1/a/1025: alarm_mgr: 01: a:00 Major Chassis Temperature
above 55 degrees C (131 F) threshold

```

zSH> alarm show

```

*****      Central Alarm Manager      *****
      ActiveAlarmCurrentCount           :21
      AlarmTotalCount                   :101
      ClearAlarmTotalCount              :80
      OverflowAlarmTableCount           :0

ResourceId      AlarmType      AlarmSeverity
-----
1-a-3-0/eth     linkDown      critical
1-a-4-0/eth     linkDown      critical
1-a-5-0/eth     linkDown      critical
1-a-6-0/eth     linkDown      critical
1-a-7-0/eth     linkDown      critical

```

1-a-8-0/eth	linkDown	critical
1-a-9-0/eth	linkDown	critical
1-a-10-0/eth	linkDown	critical
1-a-11-0/eth	linkDown	critical
1-b-1-0/eth	linkDown	critical
1-b-3-0/eth	linkDown	critical
1-b-4-0/eth	linkDown	critical
1-b-5-0/eth	linkDown	critical
1-b-6-0/eth	linkDown	critical
1-b-7-0/eth	linkDown	critical
1-b-8-0/eth	linkDown	critical
1-b-9-0/eth	linkDown	critical
1-b-10-0/eth	linkDown	critical
1-b-11-0/eth	linkDown	critical
system	temp_over_limit	major
slot 13	card_removed	warning

View the alarm when the outlet temperature exceeds the configured temperature high threshold by +10.

```
zSH> JUL 28 10:07:58: alert : 1/a/12 : shelfctrl: Warning: Temperature is above
60 degrees C (140 F) threshold.
JUL 28 10:07:58: alert : 1/a/12 : shelfctrl: Outlet temp=60 degrees C (140 F)
JUL 28 10:07:58: alert : 1/a/1025: alarm_mgr: 01: a:00 Minor Updating
Temperature alarm severity
JUL 28 10:07:58: alert : 1/a/1025: alarm_mgr: 01: a:00 Critical Chassis
Temperature above 60 degrees C (140 F) threshold
```

zSH> **alarm show**

```
***** Central Alarm Manager *****
ActiveAlarmCurrentCount :21
AlarmTotalCount :102
ClearAlarmTotalCount :81
OverflowAlarmTableCount :0
```

ResourceId	AlarmType	AlarmSeverity
-----	-----	-----
1-a-3-0/eth	linkDown	critical
1-a-4-0/eth	linkDown	critical
1-a-5-0/eth	linkDown	critical
1-a-6-0/eth	linkDown	critical
1-a-7-0/eth	linkDown	critical
1-a-8-0/eth	linkDown	critical
1-a-9-0/eth	linkDown	critical
1-a-10-0/eth	linkDown	critical
1-a-11-0/eth	linkDown	critical
1-b-1-0/eth	linkDown	critical
1-b-3-0/eth	linkDown	critical
1-b-4-0/eth	linkDown	critical
1-b-5-0/eth	linkDown	critical
1-b-6-0/eth	linkDown	critical
1-b-7-0/eth	linkDown	critical
1-b-8-0/eth	linkDown	critical
1-b-9-0/eth	linkDown	critical
1-b-10-0/eth	linkDown	critical

1-b-11-0/eth	linkDown	critical
system	temp_over_limit	critical
slot 13	card_removed	warning

View the alarm when the outlet temperature reaches the configured temperature low threshold.

```
zSH> JUL 28 11:51:03: alert : 1/a/12 : shelfctrl: Warning: Temperature is below
0 degrees C (32 F) threshold.
JUL 28 11:51:03: alert : 1/a/12 : shelfctrl: Outlet temp=0 degrees C (32 F)
JUL 28 11:51:03: alert : 1/a/1025: alarm_mgr: 01: a:00 Minor Chassis Temperature
below 0 degrees C (32 F) threshold
```

zSH> **alarm show**

```
***** Central Alarm Manager *****
ActiveAlarmCurrentCount :21
AlarmTotalCount :112
ClearAlarmTotalCount :91
OverflowAlarmTableCount :0
```

ResourceId	AlarmType	AlarmSeverity
-----	-----	-----
1-a-3-0/eth	linkDown	critical
1-a-4-0/eth	linkDown	critical
1-a-5-0/eth	linkDown	critical
1-a-6-0/eth	linkDown	critical
1-a-7-0/eth	linkDown	critical
1-a-8-0/eth	linkDown	critical
1-a-9-0/eth	linkDown	critical
1-a-10-0/eth	linkDown	critical
1-a-11-0/eth	linkDown	critical
1-b-1-0/eth	linkDown	critical
1-b-3-0/eth	linkDown	critical
1-b-4-0/eth	linkDown	critical
1-b-5-0/eth	linkDown	critical
1-b-6-0/eth	linkDown	critical
1-b-7-0/eth	linkDown	critical
1-b-8-0/eth	linkDown	critical
1-b-9-0/eth	linkDown	critical
1-b-10-0/eth	linkDown	critical
1-b-11-0/eth	linkDown	critical
system	temp_under_limit	minor
slot 13	card_removed	warning

MXK card configuration

This section describes how to provision MXK cards:

- [View uplink cards, page 148](#)
- [View line cards, page 148](#)
- [MXK card configuration, page 149](#)

View uplink cards

You can view information by entering the **slots** command with the uplink card slot of the uplink card including:

- ROM Version
- Software Version
- Card-Profile ID

The asterisk next to the type of card indicates that this card is in a redundant configuration.

```
zSH> slots a
MXK 819
Type          : *MXK TWO TENGIGE EIGHT GIGE
Card Version  : 800-02485-01-A
EEPROM Version : 1
Serial #      : 1360640
CLEI Code     : No CLEI
Card-Profile ID : 1/a/10100
Shelf         : 1
Slot          : a
ROM Version   : MXK 2.0.100
Software Version: MXK 2.5.1.124
State         : RUNNING
Mode          : FUNCTIONAL
Heartbeat check : enabled
Heartbeat last  : TUE MAR 11 18:55:46 2014
Heartbeat resp  : 4243
Heartbeat late  : 0
Hbeat seq error : 0
Hbeat longest   : 5
Fault reset     : enabled
Power fault mon : not supported
Uptime         : 3 days, 1 hour, 31 minutes
```

View line cards

After you install the uplink card in slot *a*, all other line cards and the uplink card in slot *b* (for redundant configurations) must be provisioned.

The **slots** command shows the cards currently exist in the MXK chassis and their state including: running, loading, not provisioned, booting, and configuring.

```
zSH> slots
MXK 819
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
  1: MXK ADSL-48-A Bonded/with 900 Ohm Splitter (RUNNING)
  4: MXK 20 ACT ETH (RUNNING)
  5: MXK 8 PORT GPON (RUNNING)
  6: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
 11: MXK 4 PORT GPON (RUNNING)
 14: MXK 20 ACT ETH (RUNNING)
 17: MXK 24 PORT VDSL2 POTS (NOT_PROV)
 18:*MTAC RING (RUNNING)
```

Enter the **slots slot number** command to display particular card information. In this case, entering **slots 10** displays information about the line card in slot 6. You can find the ROM, software version, and other card information.

```
zSH> slots 6
MXK 819
Type           : MXK 20 ACT ETH SINGLE SLOT
Card Version   : 800-03010-01-A
EEPROM Version : 1
Serial #       : 4262620
CLEI Code      : No CLEI
Card-Profile ID : 1/6/10207
Shelf          : 1
Slot           : 6
ROM Version    : MXK 2.0.100
Software Version: MXK 2.5.1.124
State          : RUNNING
Mode           : FUNCTIONAL
Heartbeat check : enabled
Heartbeat last  : TUE MAR 11 18:57:42 2014
Heartbeat resp  : 4283
Heartbeat late  : 0
Hbeat seq error : 0
Hbeat longest   : 13
Fault reset     : enabled
Power fault mon : not supported
Uptime         : 27 days, 17 hours, 30 minutes
```

MXK card configuration

This section describes how to:

- [Add a card profile, page 150](#)
- [Delete a card profile, page 151](#)

- [Add a card that returns parameter prompts, page 152](#)
- [card stats command, page 155](#)

Add a card profile

The MXK distinguishes the differences between cards and their functionality by designating a card type with the **card add** command.

To provision the cards in a MXK chassis enter **card add slotnumber**. This command automatically creates the **card-profile** for the card. The slot number determines the card type.

Adding a card profile

If necessary, use the **slots** command to verify which slot a card resides in before using the **card add** command to provision the card. To provision a card, first install the card in a slot.

- 1 To verify the location of a card, enter **slots**:

```
zSH> slots
MXK 819
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
  1:*TAC ITM RING (RUNNING)
  5: MXK 8 PORT GPON (RUNNING)
  6: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
  7: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
  8: MXK ADSL-48-A Bonded/with 600 Ohm Splitter (RUNNING)
 10: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (NOT_PROV)
 12: MXK 24 PORT VDSL2 (RUNNING)
```

- 2 To provision a card, enter **card add slotnumber**:

```
zSH> card add 10
card-profile validation failed - card-line-type not compatible with card
sub-type
```

In this case, the MXK-ADSL-48-A Bonded/ with Packet Voice POTS, RNG, ITM card needs to have the **card-line-type** designated.

The correct **card-line-type** for the MXK-ADSL-48-A Bonded/ with Packet Voice POTS, RNG, ITM card is *adsl-pots-pv-rng-itm*. See [Add a card that returns parameter prompts on page 152](#) for more information on line card types.

Enter **card add slotnumber linetype** type:

```
zSH> card add 10 linetype adsl-pots-pv-rng-itm
new card-profile 1/10/10202 added, sw-file-name "mxlc48aadslbond.bin", 1
option: card-line-type adsl-pots-pv-rng-itm
```

3 To verify the state of the provisioning, enter **slots** again:

```
zSH> slots
MXK 819
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
  1:*TAC ITM RING (RUNNING)
  5: MXK 8 PORT GPON (RUNNING)
  6: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
  7: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
  8: MXK ADSL-48-A Bonded/with 600 Ohm Splitter (RUNNING)
 10: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (LOADING)
 12: MXK 24 PORT VDSL2 (RUNNING)
```

After a bit, verify the state of the card again.

```
zSH> slots
MXK 819
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
  1:*TAC ITM RING (RUNNING)
  5: MXK 8 PORT GPON (RUNNING)
  6: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
  7: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
  8: MXK ADSL-48-A Bonded/with 600 Ohm Splitter (RUNNING)
 10: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (RUNNING)
 12: MXK 24 PORT VDSL2 (RUNNING)
```

Delete a card profile

Deleting a card, deletes the **card-profile** interface and all provisioning including any associated routing **ip-interface-record** profiles and bridging **bridge-interface-record** profiles.

Deleting a card profile



Caution: Before deleting card profiles, perform the following:

- Back up the MXK configuration. See the release notes for information.
- For voice cards, ensure all subscribers and voice profiles are deleted before deleting the card.
- Remove the card from the system as explained in the *MXK Hardware Installation Guide*.

Delete the **card-profile** for a card to delete all the profiles associated with a card. After deleting the card, the specified card reboots.

The **card delete** command uses the following syntax:

card delete shelf/slot/cardtype

```
zSH> card delete 1/13/10200
card-profile 1/13/10200 deleted
zSH> JUN 29 16:15:35: critical: 1/13/1035: rebootserver:
* * * * Slot Reboot : type = 2, shelf = 1, slot = 13
JUN 29 16:15:34: info    : 1/a/1054: carddeletehdlr: Starting residual
profile deletions for card 1/13/10200
JUN 29 16:16:09: info    : 1/a/1054: carddeletehdlr: Residual profile
deletions in progress for card 1/13 (100 records removed)
JUN 29 16:16:10: info    : 1/a/1054: carddeletehdlr: Completed residual
profile deletions for card 1/13/10200 (113 records removed)
```

The following slots commands show the change of status of the Active Ethernet card in slot 1 immediately after entering **card delete**. The state of the card changes from running to not provisioned.

```
zSH> slots
MXK 819
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (NOT_PROV)
Cards
  9: MXK 4 PORT GPON (NOT_PROV)
 13: MXK 20 ACT ETH (RUNNING)
```

The system also displays a message that all provisioning associated with the card is being deleted.

```
zSH> slots
MXK 819
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
  4: MXK 4 PORT GPON (RUNNING)
 13: MXK 20 ACT ETH (NOT_PROV)
```



Note: You can only delete one card at a time. Wildcards are not supported when deleting cards and their profiles.

Add a card that returns parameter prompts

There are several cards for the MXK where you must enter a valid *variable* for the **card-line-type** parameter. To view the default variables for the **card-profile** profile, enter:

```
zSH> show card-profile
sw-file-name:-----> {68}
admin-status:-----> operational  disable  maintenance  warmreset  reset
upgrade-sw-file-name:---> {68}
```

```

upgrade-vers:-----> {36}
admin-status-enable:----> enable disable
sw-upgrade-admin:-----> loadupgradesw upgradenow upgradeonreset reloadcurrrev
sw-enable:-----> true false
sw-upgrade-enable:-----> true false
card-group-id:-----> {0 - 0}
hold-active:-----> true false
weight:-----> neveractive nopreference slightpreference
mediumpreference highpreference
card-line-type:-----> unknowntype e1 ds1 e1-ima ds1-ima e3 ds3
t1-uni-gr303 t1-ima-gr303 e1-uni-v52 e1-ima-v52 gshdsl t1-uni-tlcas t1
-ima-tlcas tlcas rpr rpr-t1-gr303 rpr-e1-v52 rpr-tlcas adsl-pots adsl-pots-pv
adsl-splitter adsl-pots-pv-rng-itm ebs ebs-pv ebs-pots-pv pot s pots-pv isdn
isd-pv pots-coin pots-coin-pv reach-splitter t1-tr008 gshdsl-ntp gshdsl-nt
card-atm-configuration:-> notapplicable cellrelayonly cellrelayandmanagement
dataterm voicegateway hybridlowaal5data hybriddefault hybridhighaa l5data
vbnrt95rt5 vbnrt80rt15 vbnrt65rt30 vbnrt50rt45 vbnrt35rt60 vbnrt20rt75
vbnrt5rt95 vbnrt5rt95cbr
card-line-voltage:-----> not-used 60-volts 68-volts 95-volts 100-volts
110-volts
maxvpi-maxvci:-----> notapplicable vpi15-vci63 vpi7-vci127 vpi15-vci127
card-init-string:-----> {256}
wetting-current:-----> disabled standard
pwe-timing-mode:-----> none source-differential source-adaptive
remote-differential remote-adaptive

```

In the case of a MXK TAC card, there are two parameters that must be set. A prompt will return for each of the parameters even when the first parameter is designated. For example:

```

zSH> card add 1
card-group-id validation failed - card-group-id is 0
      use "group" option to set card-group-id
zSH> card add 1 group 2
card-profile validation failed - card-line-type must be either e1 or ds1

```

The **card add** command must be entered with all of the parameter variables designated.

```

zSH> card add 1 linetype ds1 group 2
An autogenerated card-group-id [2] is assigned for this card type.
new card-profile 1/1/5072 added, sw-file-name "tacitmring.bin", 2 options:
card-group-id 2 card-line-type ds1

```

Table 13: Card configuration

Card model number	Binary image	Parameter
MXK-UPLINK-2X10G-8X1GE	<i>mxup2tg8g.bin</i>	defaults accepted
MXK-UPLINK-4X1GE	<i>mxup4g.bin</i>	
MXK-UPLINK-4X1GE-CU	<i>mxup4gcopper.bin</i>	
MXK-UPLINK-8X1GE	<i>mxup8g.bin</i>	
MXK-UPLINK-6X1GE-CLK	<i>mxup6g.bin</i>	
MXK-UPLINK-2X10G-8X1GE-CLK	<i>mxup2tg8gtop.bin</i>	
MXK-UPLINK-2X10G-8X1G-TOP	<i>mxup2tg8gtop.bin</i>	
MXK-GPONX4-IO	<i>mxlc8gp.bin</i>	defaults accepted
MXK-GPONX8-IO	<i>mxlc4gp.bin</i>	
MXK-ADSL2+-BCM-48A	<i>mxlc48aadslbond.bin</i>	defaults accepted
MXK-ADSL2+-BCM-48B	<i>mxlc48badslbond.bin</i>	
MXK-ADSL2+-POTS-BCM-48A-2S		
MXK-ADSL2+-SPLTR600-BCM-48A-2S		
MXK-ADSL2+-SPLTR900-BCM-48A-2S		
MXK-ADSL2+-POTS-BCM-48A-RNG-2S	<i>mxlc48aadslbond.bin</i>	linetype <i>adsl-pots-pv</i> (enter this value when using a TAC card for lookout testing) <i>adsl-pots-pv-rng-itm</i> (enter this value for lookout testing without a TAC card)
MXK-EFM-SHDSL-24-NTP	<i>mxlc24gshdslbond.bin</i>	defaults accepted
MXK-EFM-SHDSL-24-NTWC		
MXK-AEX20-FE/GE-2S	<i>mxlc20ae.bin</i>	defaults accepted
MXK-AEX20-FE/GE	<i>mxlc20ae1s.bin</i>	
MXK-AEX20-FE/GE-CSFP	<i>mxlc20ae1scsfp.bin</i>	
MXK-AE-2X10G-8XGE	<i>mxlc2tg8gae.bin</i>	
MXK-VDSL2-24	<i>mxlc24vdsl2.bin</i>	defaults accepted
MXK-VDSL2-SPLTR600-BCM-17A-24	<i>mxlc24vdsl2.bin</i>	
MXK-VDSL2-SPLTR900-BCM-17A-24	<i>mxlc24vdsl2.bin</i>	
MXK-VDSL2-POTS-BCM-17A-24	<i>mxlc24vdsl2pots.bin</i>	
MXK-VDSL2-POTS-BCM-17A-48	<i>mxlc48vdsl2.bin</i>	defaults accepted
MXK-EFM-T1/E1-24	<i>mxlc24t1e1bond.bin</i>	linetype <i>ds1</i> for T1 or <i>e1</i> for E1
MXK-PWE-T1/E1-24	<i>mxlc24t1e1bond.bin</i>	linetype <i>ds1</i> for T1 or <i>e1</i> for E1
MXK-POTS-72	<i>mxlc72pots.bin</i>	linetype <i>pots-pv</i>
MXK-POTS-EBS-PKT-24	<i>mxlc24ulcs.bin</i>	<i>ebs-pots-pv</i>

Table 13: Card configuration (Continued)

Card model number	Binary image	Parameter
MXK-ADSL-72	<i>mxlc72adslbond.bin</i>	defaults accepted
MXK-OC-3/STM-1 PWE	<i>mxlcoc3stm1pwe.bin</i>	linetype ds1 for T1 or e1 for E1
MXK-MTAC-ITM-RING	<i>tacitmring.bin</i>	linetype e1 or ds1 group: group number

card stats command

The **card stats** command displays runtime statistics for the MXK device.

```
zSH> card stats
----- cpu % utilization ----- memory (KB)----- Card Memory      uptime
slot idle usage high  services framework low  % Used Total Peak Avail Status      ddd:hh:mm:ss  s/w version
=====
a*  91    9    4    4          0    0   20.75 624080 129577 494589 1 - OK      2:03:18:59  MXK 2.5.1.124
```

The **card stats all** command displays information for all the cards.

```
zSH> card stats all
----- cpu % utilization ----- memory (KB)----- Card Memory      uptime
slot idle usage high  services framework low  % Used Total Peak Avail Status      ddd:hh:mm:ss  s/w version
=====
1   92    8    6    1          0    1   33.85 109387  37062  72359 1 - OK      2:03:20:51  MXK 2.5.1.124
6   92    8    5    2          0    0   42.53 104465  44451  60032 1 - OK      2:03:18:42  MXK 2.5.1.124
a*  91    9    4    4          0    0   20.75 624080 129577 494589 1 - OK      2:03:22:11  MXK 2.5.1.124
b   91    9    4    4          0    0   20.29 624081 126648 497482 1 - OK      2:03:18:34  MXK 2.5.1.124
```

Table 14: card stats command fields

Section	Field
CPU % utilization	slot Textual description of the unit/card or access device type.
	idle Percentage of time the CPU has spent executing tasks with priority of 200 or less. Tasks with priority of 200 or less (the higher the number, the lower the priority) are considered idle tasks.
	usage Percentage of time the CPU has spent executing tasks with priority of 199 or higher
	high Percentage of time the CPU has spent executing tasks with priority of 001 to 099. High priority tasks are primarily related to packet processing and critical system monitoring.

Table 14: card stats command fields (Continued)

Section	Field
	services Percentage of time the CPU has spent executing tasks with priority of 100 to 179. Services tasks are primarily line monitoring tasks for line state and alarms.
	framework Percentage of time the CPU has spent executing tasks with priority of 180 to 199. Framework tasks are primarily database and network management system related activities such as config synch and backup.
	low Percentage of time the CPU has spent executing tasks with priority of 200 to 250
memory (KB)	Used Percentage of time the CPU has spent executing tasks with priority of 199 or higher.
	Total The amount of physical memory contained by the device/card.
	Peak The maximum physical memory that has been allocated at any time by the device/card.
	Avail The amount of physical memory that is unallocated and not in use by the device/card.
Card Memory Status	Memory status of the card sent with memory trap. A trap is sent when each condition occurs. 1 - ramMemOK less then 90% of ram is used 2 - ramMemLow more then 90% of ram is used 3 - flashMemOK enough flash for maximum database 4- flashMemLow not enough flash for maximum database 5 - flashMemOut no more flash memory, data no longer persistent
uptime ddd:hh:mm:ss	Uptime is calculated as sysUpTime - ifLastChange (assuming the device/card is running).
s/w version	Software version.

MXK DNS resolver configuration

Domain Name System (DNS) maps domain names to IP addresses, enabling the system to reach destinations when it knows only the domain name of the destination. In other words, you can use **ping** and a name instead of an IP address. DNS configuration uses the following profiles:

- **resolver**—Configures the global DNS resolver, including the DNS search order, default domain name, and list of nameserver addresses. The DNS settings in this record can be used for local applications by administrators on the system, such as **traceroute** or **ping**.
- **host-name**—A replacement for the UNIX local hosts table. Up to four host aliases can be defined for each host entry. Settings in the **resolver** record determine whether the hosts table is searched.

[Table 15](#) describes the configurable parameters for the **resolver** profile (all others should be left at their default values):

Table 15: Configurable resolver parameters

Parameter	Description
query-order	The kind of resolver query for this routing domain. Values: hosts-first searches the local hosts table first then the list of nameservers. dns-first searches the list of nameservers first then the local hosts table. dns-only searches only the list of nameservers. Default: hosts-first
domain	The routing domain to which this host parameter applies. The default is an empty string. The only routing domain supported is domain 1.
first-nameserver	The IP address of the first or primary nameserver for this routing domain. The default value is 0.0.0.0.
second-nameserver	The IP address of the second or secondary nameserver for this routing domain. This nameserver is queried if the first nameserver cannot resolve the query. The default value is 0.0.0.0.
third-nameserver	The IP address of the third or tertiary nameserver for this routing domain. This nameserver is queried if the first nameserver cannot resolve the query. The default value is 0.0.0.0.

The following example creates a **resolver** record for a routing domain:

```
zSH> new resolver 1
Please provide the following: [q]uit.
query-order: -----> {hosts-first}:
domain: -----> {}: zhone.com
first-nameserver: --> {0.0.0.0}: 192.168.8.21
```

```

second-nameserver: -> {0.0.0.0}: 201.23.20.2
third-nameserver: --> {0.0.0.0}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
Record created.

```

Another way to create DNS is by creating a **hosts** profile after the **resolver** profile is created. The syntax is **new host-name routingdomain/ipoctet1/ipoctet2/ipoctet3/ipoctet4**.

[Table 16](#) describes the configurable parameters in the **host-name** profile (all others should be left at their default values).

Table 16: Configurable parameters in the host-name profile

Parameter	Description
hostname	Client host name (if any) that the client used to acquire its address. The default is an empty string.
hostalias1	Host name alias for the specified host. The default value is an empty string.
hostalias2	Secondary host name alias for the specified host. The default value is an empty string.
hostalias3	Tertiary host name alias for the specified host. The default value is an empty string.
hostalias4	Quaternary host name alias for the specified host. The default value is an empty string.

```

zSH> new host-name 1/192/168/8/32
Please provide the following: [q]uit.
hostname: ---> {}: www.zhone.com
ipaddress: --> {0.0.0.0}: 192.168.8.32
hostalias1: -> {}: engineering.zhone.com
hostalias2: -> {}: marketing.zhone.com
hostalias3: -> {}: sales.zhone.com
hostalias4: -> {}: gss.zhone.com
.....
Save new record? [s]ave, [c]hange or [q]uit: s
Record created.

```

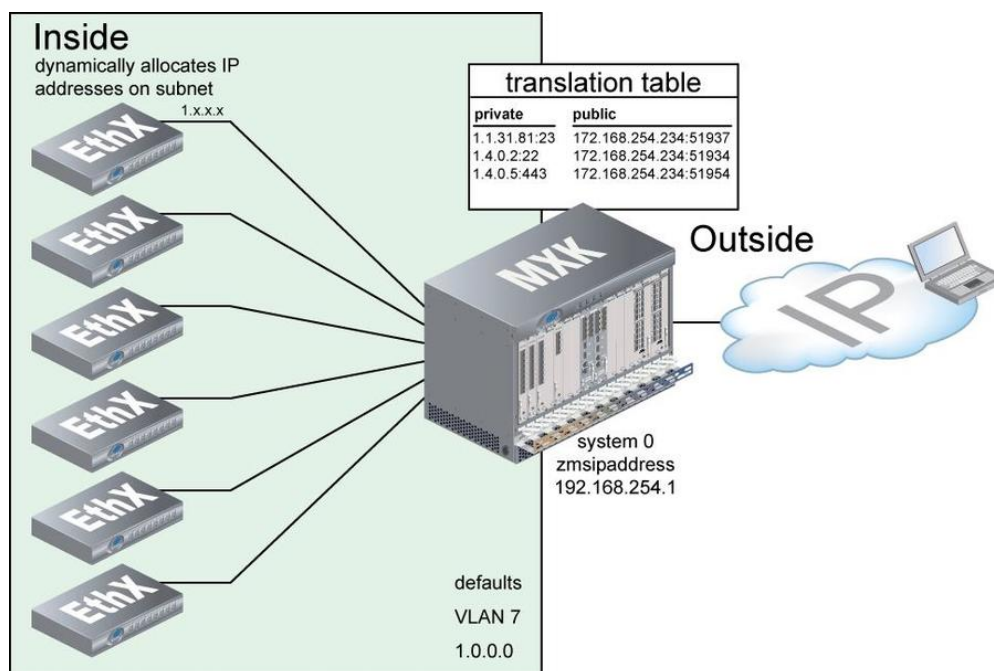
CPE Manager

The MXK's CPE Manager provides a means for managing customer premises equipment (CPE) devices without requiring extra routable IP addresses to reach these CPE end-points. While the CPE Manager is specifically designed for Zhone's EtherXtend and zNID family of CPE products, CPE Manager can be used with any CPE device which supports receiving an IP address via DHCP on a VLAN.

In many service provider networks, the increasing usage of IP-aware CPE devices creates an operational challenge for service providers because the

number of devices which require IP addresses cause IP address space depletion, making it hard to assign routable addresses for these devices.

A solution to this problem is the SLMS CPE Manager. CPE Manager adds proxy capability to SLMS, allowing one IP interface on the Zhone central office device to provide IP access to all the subtended CPE devices connected to it. This one IP interface is created on an upstream port which is routable on the service providers management network, and it provides IP address and protocol port translation when forwarding packets to and from managed CPE devices. In this way, IP can be used for CPE management without having to consume IP address space or having to add network routes for reachability of line side CPE devices.



CPE Manager is supported on the following line cards:

- MXK-EFM-SHDSL-24-NTWC
- MXK-EFM-SHDSL-24-NTP
- MXK-AEX20-FE/GE-2S
- MXK-AEX20-FE/GE
- MXK-GPONX8-IO
- MXK-GPONX4-IO

Accessing the CPE's private address, ports

To access a CPE configured using CPE Manager, access the MXK through its IP address, however, instead of using the well known protocol ports, use the CPE's base public port plus an offset to the specific port used for the protocol

desired. Supported protocols include Echo, FTP (data), FTP (control), SSH, Telnet, HTTP, SNMP and HTTPS.

To select the ports to make available the **cpe-mgr add** command has several options depending on the selection of **the compact and security parameters**:

- **compact [full | partial | none]**

Selection of the **compact** mode defines how many ports may be accessed using the NAT-PAT binding, the more ports are accessed per device, the fewer devices that will be able to be accessed.

- **security [enabled | disabled | default]**

Selection of the security mode defines whether those ports will use SSH, for example HTTP or HTTPS, telnet or SSH.

A list of offsets for public ports based on the compact and security mode is given in [Offsets for public ports, page 160](#). For more information about how offsets work, see [Additional information about CPE manager on page 167](#).

The defaults for compact mode is full mode (the three port mapping). For security mode, the default is default, which means to use the security settings for the MXK chassis in system 0. For additional information about security and system 0, see [Enable security on the MXK on page 128](#).

Table 17: Offsets for public ports

Public port offset	Type	Name	Compact & Security Modes				
			Full		Partial		None
			Secure Enabled	Secure Disabled	Secure Enabled	Secure Disabled	N/A (all ports)
7	TCP, UDP	ECHO	+0	+0	+0	+0	+0
20	TCP	FTP - data	—	—	—	—	+1
21	TCP	FTP - control	—	—	—	—	+2
22	TCP, UDP	SSH	+1	—	+1	—	+3
23	TCP, UDP	Telnet	—	+1	—	+1	+4
80	TCP	HTTP	—	+2	—	+3	+5
81	TCP	HTTP	—	—	—	—	+6
161	TCP, UDP for partial and none UDP for full compact mode	SNMP	+2	+2	+2	+2	+7
162	UDP	SNMP traps (upstream only)	+0	+0	+3	+3	+1

Table 17: Offsets for public ports

Public port offset	Type	Name	Compact & Security Modes				
			Full		Partial		None
			Secure Enabled	Secure Disabled	Secure Enabled	Secure Disabled	N/A (all ports)
443	TCP	HTTPS	+2	—	+3	—	+8

The private class A network is set up by default as 1.0.0.0/8 on VLAN 7. These defaults may be changed, see [Changing the VLAN of the local network, page 162](#).

The IP addresses given to CPEs follow the general guidelines:

<Class A network>.<Slot>.<Port number: higher order byte>.<Port number: lower order byte>

Note that the GPON format has the port/subport encoded into the IP address which allows 12 bits for a subport and 4 bits for the port number:

<class A>.<slot>.<subport upper 8 bits>.<subport lower 4 bits * 16 + port>

The *1-1-4-501/gponport* yields an IP address of 1.1.31.84.

Configuring the MXK as a CPE manager for Active Ethernet

Setting up the CPE manager from the CLI is fairly simple. First you have to have an IP address on an upstream port.

1 Add a public address for the CPE manager

```
zSH> cpe-mgr add public 192.168.254.1
CPE Manager using 192.168.254.1 for public interface.
```

Configuring the public address for the MXK requires that the MXK has already been given an IP address.

2 Add the local device to the CPE manager.

```
zSH> cpe-mgr add local 1-13-1-0/eth
Configured CPE Manager's local network:
  Class A network: 1.0.0.0
  Local IP:       1.0.0.1
  VLAN ID:        7
  Created CPE Management interface: 1-13-1-0-eth-7/ip
```

Note that the default network is created if you do not manually create the network first.

Configuring the MXK as a CPE manager for EFM-SHDSL

To create an EFM-SHDSL bond group, see [Bond group configuration, page 1412](#).

- 1 Add a public address for the CPE manager

```
cpe-mgr add public 192.168.254.1
```

- 2 Add the local device to the CPE manager.

```
cpe-mgr add local 1-3-42-0/efmbond
```

Configuring the MXK as a CPE manager for GPON

Adding CPE manager is a little different for GPON.

- 1 Add a public address for the CPE manager

```
cpe-mgr add public 192.168.254.1
```

- 2 Add a GPON zNID

The following work if the GPON port already exists.

```
cpe-mgr add local 1-11-1-501/gponport
```

If the GPON port does not exist, it can be created within the cpe-mgr add local command by adding **gtp <gpon-traffic-profile index>**:

```
zSH> cpe-mgr add local 1-1-1-501/gponport gtp 1
GEM Port 1-1-1-501/gponport has been created on ONU
1-1-1-1/gpononu.
Created CPE Management interface:
1-1-1-501-gponport-7/ip
```

Changing the VLAN of the local network

Ordinarily the default settings are acceptable. However if you need to change the default class A network or VLAN ID you can use the following command, however you should not that if you change the VLAN you would need to change the VLAN settings of all the CPEs. VLAN 7 is the default management VLAN setting of Zhone zNIDs and EtherXtend devices.

To change the VLAN ID for the CPE manager local private network

```
cpe-mgr add local vlan <vlan id to use internally for management>
```

If you were to manually set the VLAN ID to the default, you would use

```
cpe-mgr add local vlan 7
```



Note: Zhone does not recommend changing the VLAN manually because Zhone CPE and zNID products use VLAN 7 as the default management VLAN.

Changing the class A network used as the CPE manager local network

Once again the default settings should be acceptable. However if you need to change the default class A network the following command may be used. If you want to change network settings after CPEs are attached and configured you would have to delete them all before making the changes:

To manually set the local network settings

```
cpe-mgr add local network <class A network used internally for all managed CPEs>
```

If you were to manually set the local network to the default, you would use

```
cpe-mgr add local network 1.0.0.0
```



Note: You can only manually set the local network settings when no CPE devices are currently configured on the network.

By default we use the 1.0.0.0 class A network. In other words, a class A network is one that has an 8 bit mask which means only the first byte of the IP address is common between nodes in the network. If you execute the following command: **cpe-mgr add local network 2.0.0.0**, the class A network will be changed and all local IP will start with 2.

Viewing the CPE Manager ports

The **cpe-mgr show** command provides a mapping between the interface and the local IP address along with the various ports. For more information on available ports see [Additional information about CPE manager, page 167](#).

```
zSH> cpe-mgr show          CPE Manager public side
interface:
```

```
IP:          192.168.254.234
```

```
ifIndex: 73
```

```
CPE Manager local management network:
```

```
IP:          1.0.0.1/8 (default) (active)
```

```
VlanID: 7 (default)
```

Managed CPE Interface Configuration:

InterfaceLocal IPECHOFTPSSHTelntHTTPSNMPHTTPS

```

-----
-----
1-4-9-0/eth1.4.0.951921 - -519225192351923 -
1-7-41-0/efmbond1.7.0.4151924 - -519255192651926 -
1-1-4-501/gponport1.1.31.8451927 - -519285192951929 -
1-4-1-0/eth1.4.0.151930 - -519315193251932 -
1-1-1-501/gponport1.1.31.8151936 - -519375193851938 -
1-4-2-0/eth1.4.0.251933 -51934 - -5193551935
1-4-3-0/eth1.4.0.351939519405194251943519445194651947
1-4-4-0/eth1.4.0.451948 - -519495195151950 -
1-4-5-0/eth1.4.0.551952 -51953 - -5195551954

```

Compact mode full with security disabled.

```

zSH> cpe-mgr show local 1-1-1-501/gponport
      Public IP address: 192.168.254.234
      Public Access Port:
                Protocol      Port
                ECHO          51936
                SNMP Traps    51936
                Telnet        51937
                HTTP          51938
                SNMP          51938
      Local IP Address: 1.1.31.81

```

Compact mode full with security enabled.

```

zSH> cpe-mgr show local 1-4-2-0/eth
      Public IP address: 192.168.254.234
      Public Access Port:
                Protocol      Port
                ECHO          51933
                SNMP Traps    51933
                SSH           51934
                HTTPS         51935
                SNMP          51935
      Local IP Address: 1.4.0.2

```


Compact mode none. Note that since all ports are available security mode is not applicable in this case.

```
zSH> cpe-mgr show local 1-4-3-0/eth
Public IP address: 192.168.254.234
Public Access Port:
      Protocol      Port
      ECHO          51939
      SNMP Traps    51940
      FTP            51940/51941
      SSH            51942
      Telnet         51943
      HTTP(80)       51944
      HTTP(81)       51945
      SNMP           51946
      HTTPS          51947
Local IP Address: 1.4.0.3
```

Compact mode partial with security disabled.

```
zSH> cpe-mgr show local 1-4-4-0/eth
Public IP address: 192.168.254.234
Public Access Port:
      Protocol      Port
      ECHO          51948
      Telnet         51949
      SNMP           51950
      HTTP           51951
      SNMP Traps    51951
Local IP Address: 1.4.0.4
```

Compact mode partial with security enabled.

```
zSH> cpe-mgr show local 1-4-5-0/eth
Public IP address: 192.168.254.234
Public Access Port:
      Protocol      Port
      ECHO          51952
      SSH            51953
      SNMP           51954
      HTTPS          51955
      SNMP Traps    51955
Local IP Address: 1.4.0.5
```

Troubleshooting CPE Manager

To verify or troubleshoot CPE manager, you should understand what the two commands for CPE manager do. The first **cpe-mgr add public** command

- Sets **natenabled** to “yes” in the **ip-interface-record** for the public address (in our example, the 192.168.254.1 address)

When using the defaults and the local network has not been created, the second command, **cpe-mgr add local**:

- Creates a floating **ip-interface** record with IP address of 1.0.0.1 (only created if the defaults are being used and if the record does not already exist. In other words, the first **cpe-mgr add local** if the record wasn't created manually)
- Creates an **ip-unnumbered-record** for the floating ip-interface record (only created if the defaults are being used and if the record does not already exist. In other words, the first **cpe-mgr add local** if the record wasn't created manually)
- Creates a **dhcp-server-subnet** for the 1.0.0.0 network (only created if the defaults are being used and if the record does not already exist. In other words, the first **cpe-mgr add local** if the record wasn't created manually)
- Creates a host **ip-interface-record** for the CPE on interface (in our example bond group)

Assigns a local IP address based on the interface description (not routable, but may be reached from the private local network, or by Telnet to the MXK, then Telnet from the MXK to the device)

- Creates a **pat-bind** profile of type **cpemgr** or **cpemgrsecure**



Note: The **ip-interface-record** created is not a normal “host” record and cannot be seen using the **host show** command.

The pat-bind profile for the first device from the example ([Configuring the MXK as a CPE manager for Active Ethernet on page 161](#)) contains the local IP address (1.3.0.42) and the CPE base port (51921):

```
zSH> list pat-bind
pat-bind 1
1 entry found.
zSH> get pat-bind 1
pat-bind 1
public-ipaddr: -> {192.168.254.1}
public-port: ---> {51921}
local-ipaddr: --> {1.3.0.42}
local-port: ----> {9}
portType: -----> {cpemgr}
```

The local address which is given is based on the interface in the form:

```
<local class A network>.<slot>.<port HI byte>.<port LO
byte>
```

From our example bond group, *1-3-42-0/efmbond*, the local IP address (as shown above in the **pat-bind 1** profile) is 1.3.0.42. If you need to verify this number, do a get on the **pat-bind** profile.

Note that GPON format allows 12 bits for a subport and 4 bits for the port number:

```
<class A>.<slot>.<subport upper 8 bits>.<subport lower 4
bits * 16 + port>
```

The 1-1-4-501/gponport yields an IP address of 1.1.31.84.

Additional information about CPE manager

The first device will be accessible by the MXK's public IP address and the CPE base port. The CPE base port for the first device is 51921. To reach one of the well known ports you then give the offset for the public port. Well known port (7) is for echo which has an offset of zero.

1st device	ECHO	+0	51921
	FTP (data)	+1	
	FTP (control)	+2	
	SSH	+3	
	Telnet	+4	
	HTTP	+5	
	HTTP	+6	
	SNMP	+7	
	HTTPS	+8	
2nd device	ECHO	+0	51930
	FTP (data)	+1	
	FTP (control)	+2	
	SSH	+3	
	Telnet	+4	
	HTTP	+5	
	HTTP	+6	
	SNMP	+7	
	HTTPS	+8	
3rd device	ECHO	+0	51938
	FTP (data)	+1	
	FTP (control)	+2	
	SSH	+3	
	Telnet	+4	
	HTTP	+5	
	HTTP	+6	
	SNMP	+7	
	HTTPS	+8	
•			
•			
•			



Note: The examples use compact mode none. See [Configuring the MXK as a CPE manager for Active Ethernet on page 161](#), [Configuring the MXK as a CPE manager for EFM-SHDSL on page 162](#), and [Configuring the MXK as a CPE manager for GPON on page 162](#). Using different variations of **compact** mode and **security** mode requires different offsets as shown in [Offsets for public ports, page 160](#).

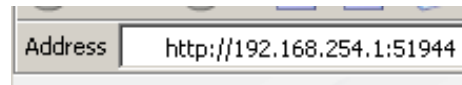
To telnet to the first CPE via the well known port, 23, you would use the CPE base port plus the public port offset of 4; You would use the MXK's address (192.168.254.1), then 51925 (51921 + 4) to Telnet to the device. From a Unix or DOS prompt it would look like

```
telnet 192.168.254.1 51925
```

To access the second device you need to start with the CPE base port for that device. Each device consumes nine public ports, so the first device has a port

range from 51921 - 51929, the second device has a port range from 51930 - 51938, the third from 51939 - 51947 and so on.

To access the HTTP port on the third device from a browser, you would start from the first public port address 51921 + 18 (the 51921 start point plus two times nine for the first two devices to get to the third device range) + 5 (to get to port 80, a HTTP port) or 51944.



As CPE devices are deleted or added, holes will form in the list of CPE devices, so the order eventually becomes arbitrary, but is used in the discussion to elucidate how the mechanism works.

CPE base port and information for added devices is shown in the cpe-mgr show display. See [Section 2, Viewing the CPE Manager ports](#).

Web UI cut-through for EtherXtend devices

This section provides the configuration procedure to create hyperlinks in the MXK Web UI that when clicked, will take you to the Web UI for the EtherXtend 3400. See [Figure 8](#) and [Figure 9](#).

Creating a Web UI cut-through for EtherXtend devices

From the MXK CLI:

- 1 Create a management interface for the MXK.
- 2 Create a CPE public IP using the MXK management IP.

```
zSH> cpe-mgr add public 172.24.200.163
CPE Manager using 172.24.200.163 for public interface.
```

- 3 Create an EFM bond group, then add the links.

```
zSH> bond add group 1-1-25-0/efmbond
Group ID {25} is already in use.
Bond group - bond-0032/efmbond - was successfully created.
```

```
zSH> bond add member bond-0032/efmbond 1-1-1-0/shdsl
```

```
zSH> bond add member bond-0032/efmbond 1-1-2-0/shdsl
```

```
zSH> bond add member bond-0032/efmbond 1-1-3-0/shdsl
```

```
zSH> bond add member bond-0032/efmbond 1-1-4-0/shdsl
```

- 4 Create a local cpe-mgr IP for the bond group.

```
zSH> cpe-mgr add local bond-0032/efmbond
Created CPE Management interface: bond-0032-efmbond-7/ip
```

5 View the **pat-bind** record that was automatically created.

```
zSH> get pat-bind *
public-port: ---> {51930}
local-ipaddr: --> {1.1.0.32}
local-port: ----> {9}
portType: -----> {cpemgr}
pat-bind 1
public-ipaddr: -> {172.24.200.163}
```

6 Verify the bond group **cpe-mgr** IP interface is UP.

```
zSH> interface show
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/1/1/0/ip     UP      1 172.24.200.163/24 00:01:47:1a:db:0f   ethernet1-1-200
1/1/32/0/ip    UP      1 [1.0.0.1] 1.1.0.32          bond-0032-efmbond-7
```

- 7 From a browser, launch a Web UI to the MXK management interface, <http://172.24.200.163>.
- 8 Through the WebUI, view the CPE Cut-Through URL by clicking to *Status->Service->CPE->CPE IP Hosts*.

Figure 6: The URLs for EtherXtend 3400 devices

STATUS / CPE IP Hosts

Public IP Address	Public Port	Local IP Address	BondGroup Id	CPE URL
172.24.200.170	51921	1.1.0.25		http://172.24.200.170:51926
172.24.200.170	51930	1.1.0.26		http://172.24.200.170:51935
172.24.200.170	51939	1.1.0.27		http://172.24.200.170:51944

Refresh

- 9 Click on the CPE URL to launch the WebUI for the EtherXtend 3400.

Figure 7: Web UI page for the ExtherXtend 3400



Z H O N E

ETHX-3400

The EtherXtend 3400 Series of Ethernet Access Devices (EADs) are high capacity devices capable of bonding up to eight (8) SHDSL bis copper lines for up to 45.6 Mbps of bandwidth. EtherXtend 3400 EADs supports 802.3ah EFM interface and EFM-OAM standards, as well as pre-standards based, Net-2-Net Ethernet over copper loop bonding protocols. They are intelligent demarc devices with bridging or Layer 3 routing options supported.

The EtherXtend 3400 provides dual-mode operations that allows the EtherXtend to connect to standards-based products such as the MALC broadband loop carrier, and other EtherXtend units. EtherXtend can also connect into pre-standard Ethernet loop bonding products such as Zhone's Ethernet access products.

Web Interface Login

User Name

Password

ETHX-3400



Web UI cut-through for EtherXtend devices

This section provides the configuration procedure to create hyperlinks in the MXK Web UI that when clicked, will take you to the Web UI for the EtherXtend 3400. See [Figure 8](#) and [Figure 9](#).

Creating a Web UI cut-through for EtherXtend devices

From the MXK CLI:

- 1 Create a management interface for the MXK.
- 2 Create a CPE public IP using the MXK management IP.

```
zSH> cpe-mgr add public 172.24.200.163
CPE Manager using 172.24.200.163 for public interface.
```

- 3 Create an EFM bond group, then add the links.

```
zSH> bond add group 1-1-25-0/efmbond
Group ID {25} is already in use.
Bond group - bond-0032/efmbond - was successfully created.
```

```
zSH> bond add member bond-0032/efmbond 1-1-1-0/shdsl
```

```
zSH> bond add member bond-0032/efmbond 1-1-2-0/shdsl
```

```
zSH> bond add member bond-0032/efmbond 1-1-3-0/shdsl
```

```
zSH> bond add member bond-0032/efmbond 1-1-4-0/shdsl
```

4 Create a local **cpe-mgr** IP for the bond group.

```
zSH> cpe-mgr add local bond-0032/efmbond
Created CPE Management interface: bond-0032-efmbond-7/ip
```

5 View the **pat-bind** record that was automatically created.

```
zSH> get pat-bind *
public-port: ---> {51930}
local-ipaddr: --> {1.1.0.32}
local-port: ----> {9}
portType: -----> {cpemgr}
pat-bind 1
public-ipaddr: -> {172.24.200.163}
```

6 Verify the bond group **cpe-mgr** IP interface is UP.

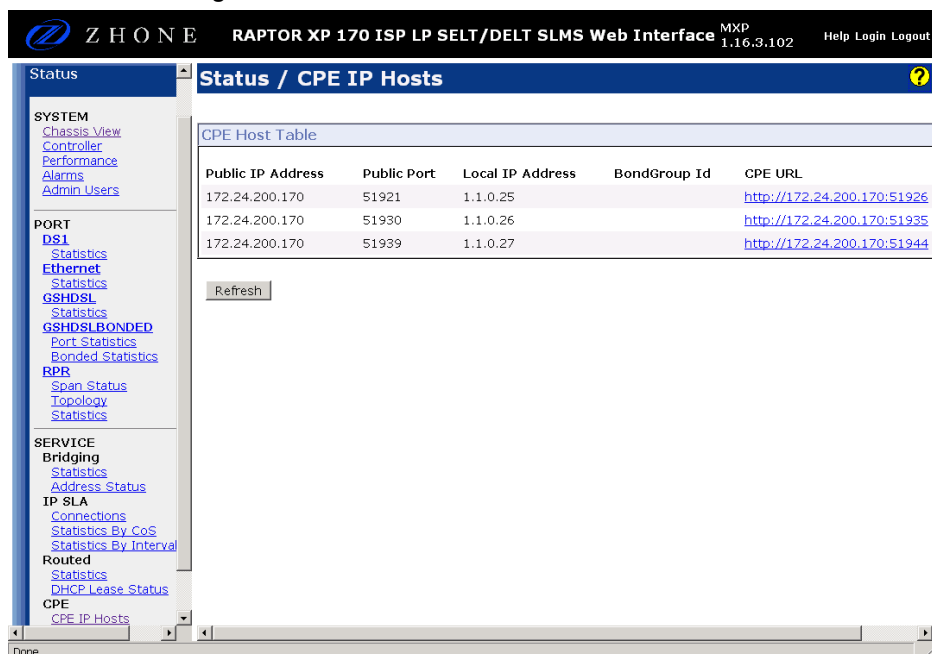
```
zSH> interface show
```

Interface	Status	Rd/Address	Media/Dest Address	IfName
1/1/1/0/ip	UP	1 172.24.200.163/24	00:01:47:1a:db:0f	ethernet1-1-200
1/1/32/0/ip	UP	1 [1.0.0.1] 1.1.0.32		bond-0032-efmbond-7

7 From a browser, launch a Web UI to the MXK management interface, <http://172.24.200.163>.

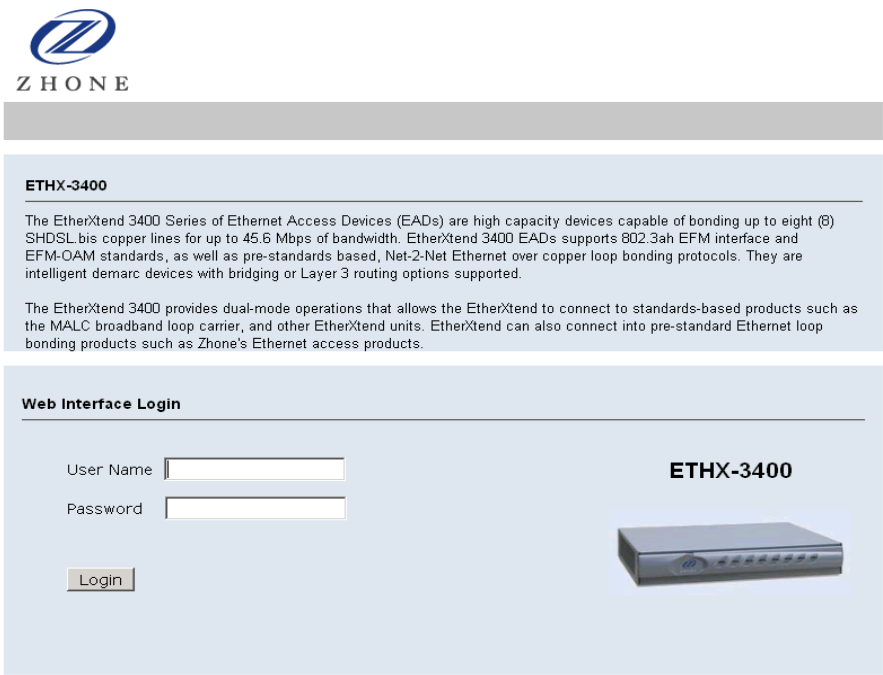
8 Through the WebUI, view the CPE Cut-Through URL by clicking to *Status->Service->CPE->CPE IP Hosts*.

Figure 8: The URLs for EtherXtend 3400 devices



9 Click on the CPE URL to launch the WebUI for the EtherXtend 3400.

Figure 9: Web UI page for the ExtherXtend 3400



3

MXK CLOCKING

This chapter describes:

- [Clock management on the MXK overview1, page 173](#)
- [MXK local and system clocking, page 174](#)
- [Set MXK system clocking from MXK sources, page 177](#)
- [Precision Time Protocol \(PTP\) and SyncE clock management on the MXK, page 184](#)

Clock management on the MXK overview1

The MXK supports five types of clocking management:

- MXK as local clocking source
See [Local clocking source on the MXK on page 174](#)
- MXK as system source for clock
Use MXK uplink or line cards as system clocking source.
 - Building Integrated Timing Source (BITS)
Special cable required. Configure line See [Set MXK system clocking from MXK sources on page 177](#).
 - T1/E1 integrated data circuits
See [Set MXK system clocking from MXK sources on page 177](#).
- Precision Time Protocol (PTP)
Clocking in master and client environment sending precision timing protocol message packets using the IEEE 1588v2 protocol.
Use the MXK-UPLINK-2X10G-8X1G-TOP only.
See [Ordinary clock and boundary clock PTP configurations, page 184](#).
- Synchronous Ethernet (SyncE)
Use the MXK-UPLINK-2X10G-8X1G-TOP only.
Ethernet IP packet timing protocol for port-to-port clock synchronization over the network.
Use the MXK-UPLINK-2X10G-8X1G-TOP only.

See [SyncE clock management, page 198](#).

MXK local and system clocking

This section describes local, and synchronized network clocking on the MXK:

- [Local clocking source on the MXK, page 174](#)
- [System clocking on the MXK overview, page 174](#)

Local clocking source on the MXK

Local clocking on the MXK is provided by the active uplink card.

System clocking on the MXK overview

When a timing source on the MXK is required, the following cards are available:

- TAC card
- T1/E1 PWE card
- EFM T1/E1 card
- 6x1GE-CLK uplink card
- 2X10G-8X1GE-CLK uplink card
- 2X10G-8X1G-TOP uplink card

To view current source of clocking on the MXK, enter **clkmgrshow**. In this case, timing is local from the uplink card.

```
zSH> clkmgrshow
All lines are using LOCAL clock
```

In this case, timing is synchronized network timing from the TAC card.

```
zSH> clkmgrshow
Primary system clock is 1/14/1/0 : T1
Secondary system clock is LOCAL timing
```

In this case, timing is synchronized network timing from the MXK 6X1GE-CLK uplink card.

```
zSH> clkmgrshow
Primary system clock is 1/30/1/0 : T1
Secondary system clock is LOCAL timing
```

To view available timing on the MXK, enter **clkmgrshow list**.

In this case, only local timing from the MXK-UPLINK-6X1GE-CLK uplink card is available on this MXK.

zSH> **slots**

MXK 823

Uplinks

a: *MXK SIX GIGE (RUNNING+TRAFFIC)

b: MXK SIX GIGE (RUNNING+TRAFFIC)

Cards

```

1: MXK 24 PORT VDSL2 POTS (RUNNING)
2: MXK 24 PORT VDSL2 POTS (RUNNING)
3: MXK 24 PORT VDSL2 POTS (RUNNING)
4: MXK 24 PORT VDSL2 POTS (RUNNING)
5: MXK 24 PORT VDSL2 (RUNNING)
7: MXK 24 PORT VDSL2 POTS (RUNNING)
8: MXK ADSL-48-A Bonded (RUNNING)
9: MXK 24 PORT VDSL2 POTS (RUNNING)
10: MXK 24 PORT VDSL2 POTS (RUNNING)
11: MXK 24 PORT VDSL2 POTS (RUNNING)
12: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (RUNNING)
14: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (RUNNING)
16: MXK T1E1-24 PWE (RUNNING)
17: MXK 8 PORT GPON (RUNNING)
18: MXK 8 PORT GPON (RUNNING)

```

zSH> **clkmgrshow list**

eligible list has 0 entries

ineligible list has 26 entries

1	not eligible	1/16/1/0	(5) :	T1 :	OOS :	LOOP
2	not eligible	1/16/2/0	(5) :	T1 :	OOS :	LOCAL
3	not eligible	1/16/3/0	(5) :	T1 :	OOS :	LOCAL
4	not eligible	1/16/4/0	(5) :	T1 :	OOS :	LOCAL
5	not eligible	1/16/5/0	(5) :	T1 :	OOS :	LOCAL
6	not eligible	1/16/6/0	(5) :	T1 :	OOS :	LOCAL
7	not eligible	1/16/7/0	(5) :	T1 :	OOS :	LOCAL
8	not eligible	1/16/8/0	(5) :	T1 :	OOS :	LOCAL
9	not eligible	1/16/9/0	(5) :	T1 :	OOS :	LOCAL
10	not eligible	1/16/10/0	(5) :	T1 :	OOS :	LOCAL
11	not eligible	1/16/11/0	(5) :	T1 :	OOS :	LOCAL
12	not eligible	1/16/12/0	(5) :	T1 :	OOS :	LOCAL
13	not eligible	1/16/13/0	(5) :	T1 :	OOS :	LOCAL
14	not eligible	1/16/14/0	(5) :	T1 :	OOS :	LOCAL
15	not eligible	1/16/15/0	(5) :	T1 :	OOS :	LOCAL
16	not eligible	1/16/16/0	(5) :	T1 :	OOS :	LOCAL
17	not eligible	1/16/17/0	(5) :	T1 :	OOS :	LOCAL
18	not eligible	1/16/18/0	(5) :	T1 :	OOS :	LOCAL
19	not eligible	1/16/19/0	(5) :	T1 :	OOS :	LOCAL
20	not eligible	1/16/20/0	(5) :	T1 :	OOS :	LOCAL
21	not eligible	1/16/21/0	(5) :	T1 :	OOS :	LOCAL
22	not eligible	1/16/22/0	(5) :	T1 :	OOS :	LOCAL
23	not eligible	1/16/23/0	(5) :	T1 :	OOS :	LOCAL
24	eligible	1/16/24/0	(5) :	T1 :	OOS :	LOOP
25	eligible	1/b/1/0	(5) :	T1 :	OOS :	LOOP (Standby)
26	not eligible	1/a/1/0	(5) :	T1 :	ACTIVE :	LOCAL

pending list has 0 entries

In this case, an TAC card is set to loop timing and is available for synchronized network timing network on this MXK.

```
zSH> clkmgrshow list
eligible list has 1 entry
  1 *      eligible  1/14/1/0  ( 5) :      T1 : ACTIVE :    LOOP
ineligible list has 0 entries
pending list has 0 entries
```

In this case, the an MXK with a TOP uplink card is configured for PTP clock.

```
zSH> clkmgrshow list
eligible list has 2 entries
  1 *      eligible  1/a/1/0    (11) :      T1 : ACTIVE :    LOOP
  2        eligible  1/a/1/0    ( 8) :      PTP : ACTIVE : UNKNOWN
ineligible list has 94 entries
  1  not eligible  1/a/2/0    ( 5) :  ETHERNET : OOS : NONE
  2  not eligible  1/a/3/0    ( 5) :  ETHERNET : OOS : NONE
  3  not eligible  1/a/4/0    ( 5) :  ETHERNET : OOS : NONE
  4  not eligible  1/a/5/0    ( 5) :  ETHERNET : ACTIVE : NONE
  5  not eligible  1/a/6/0    ( 5) :  ETHERNET : OOS : NONE
  6  not eligible  1/a/7/0    ( 5) :  ETHERNET : OOS : NONE
  7  not eligible  1/a/8/0    ( 5) :  ETHERNET : OOS : NONE
  8  not eligible  1/a/9/0    ( 5) :  ETHERNET : OOS : NONE
  9  not eligible  1/a/10/0   ( 5) :  ETHERNET : OOS : NONE
 10 not eligible  1/a/11/0   ( 5) :  ETHERNET : OOS : NONE
 11 not eligible  1/1/1/0    ( 5) :      T1 : OOS : THROUGH
 12 not eligible  1/1/2/0    ( 5) :      T1 : OOS : THROUGH
 13 not eligible  1/1/3/0    ( 5) :      T1 : OOS : THROUGH
 14 not eligible  1/1/4/0    ( 5) :      T1 : OOS : THROUGH
 15 not eligible  1/1/5/0    ( 5) :      T1 : OOS : THROUGH
 16 not eligible  1/1/6/0    ( 5) :      T1 : OOS : THROUGH
 17 not eligible  1/1/7/0    ( 5) :      T1 : OOS : THROUGH
 18 not eligible  1/1/8/0    ( 5) :      T1 : OOS : THROUGH
...
 90 not eligible  1/1/80/0   ( 5) :      T1 : OOS : THROUGH
 91 not eligible  1/1/81/0   ( 5) :      T1 : OOS : THROUGH
 92 not eligible  1/1/82/0   ( 5) :      T1 : OOS : THROUGH
 93 not eligible  1/1/83/0   ( 5) :      T1 : OOS : THROUGH
 94 not eligible  1/1/84/0   ( 5) :      T1 : OOS : THROUGH
pending list has 61 entries
BITS clock is not present
```

Set MXK system clocking from MXK sources

This section describes MXK system clocking:

- [MXK system clocking, page 177](#)
- [system-clock-profile overview, page 177](#)
- [Configure MXK line and uplink cards for system clocking, page 180](#)

MXK system clocking

The MXK can receive system clocking from one of the following sources:

- The Ds1 interfaces on the T1/E1 EFM card. (MXK-EFM-T1/E1-24)
Provides T1/E1 only, not BITS.
- The Ds1 interfaces on the PWE card. (MXK-PWE-T1/E1-24)
Provides T1/E1 only, not BITS.
- Ds1 interfaces on the TAC card. (MXK-TAC-ITM-RING)
Provides T1/E1 and BITS. BITS clock source has a type of Ds1.
- The CLK and TOP uplink card. (MXK-UPLINK-6X1GE-CLK and MXK-UPLINK-2X10G-8X1G-TOP)
Provides T1/E1 and BITS.
 - T1/E1 Ds1 interfaces.
 - Ds1 interface for BITS recognizes the cable for BITS.



Note: Interfaces that are designated as eligible clock sources must be set to **looptiming**.

system-clock-profile overview

The MXK creates a **system-clock-profile** for each interface that can provide clock for the system. These profiles define the clock sources that are eligible to provide system clock and defines the weights for the clock on the interface.

If there are multiple active interfaces configured as eligible clock sources, the system selects a clock source based on the weight configured in the **system-clock-profile**. If a primary clock source has been configured in the **system 0** profile, this clock source overrides all other clocks.

Note the following information about redundant clock sources on the MXK:

- By default, only when the card becomes the active interface is it eligible to provide clock, redundant interfaces are not eligible.
- The clock source with the highest weight becomes the primary clock source. Weights are from 1 (lowest priority) to 10 (highest priority).

- If a clock source is defined in the **primaryclocksource** parameter in the **system** profile, that clock source takes precedence over the settings in the **system-clock-source** profiles, if any. Clock sources defined in the **system 0** profile are given a weight of 11.
- If you assign weight to a clock source that is higher than the currently active clock source, or if you assign a clock source in the **system** profile, the system will switch over to the new clock source.

Table 18 describes the parameters used to provide clocking for the system.

Table 18: Clocking parameters

Parameter	Description
transmit-clock-source (ds1-profile)	There are three clocking options for Ds1 interfaces: Values: looptiming The recovered receive clock from the Ds1 is used as the transmit clock. localtiming A local (to the Ds1 interface) clock source is used on the Ds1 transmit signal. throughtiming The transmit Ds1 clock is derived from the recovered receive clock of another Ds1 interface. Interfaces that are designated as eligible clock sources cannot be set to through timing. Default: throughtiming
primaryclocksource (system 0 profile)	The <i>shelf-slot-port-subport/type</i> of an interface to provide clocking for the system. For the BITS clock on the TAC/Ring card, specify the address in the form <i>shelf-slot-1-0/ds1</i>.  Note: If configured, the setting in the primaryclocksource parameter overrides settings in the system-clock-profile for all interfaces that provide clocking.
system-clock-eligibility (system-clock-profile)	Specifies whether the interface is eligible to provide clocking for the system. Values: true false Default: false
system-clock-weight (system-clock-profile)	Assigns a weight to the clock source. If you assign weight to a clock source that is higher than the currently active clock source, the system will switch over to that clock source. Values: 1 to 10 1 is the lowest priority, 10 is the highest Default: 5

Configuring the Ds1 profile to looptiming for the synchronized network timing clock source

- 1 Verify that the interface that is to provide clock is up and active.
- 2 Verify the **transmit-clock-source** parameter in the **ds1-profile** is set to **looptiming**. This parameter *must* be set to **looptiming** for network timing to work.

```
zSH> update ds1-profile 1-4-1-0/ds1
ds1-profile 1-4-1-0/ds1
```

```
Please provide the following: [q]uit.
line-type: -----> {esf}:
line-code: -----> {b8zs}:
send-code: -----> {sendnocode}:
circuit-id: -----> {ds1}:
loopback-config: -----> {nolooop}:
signal-mode: -----> {none}:
fdl: -----> {fdlnone}:
dsx-line-length: -----> {dsx0}:
line-status_change-trap-enable: ---> {enabled}:
channelization: -----> {disabled}:
ds1-mode: -----> {csu}:
csu-line-length: -----> {csu00}:
clock-source-eligible: -----> {eligible}:
transmit-clock-source: -----> {throughtiming}:looptiming
cell-scramble: -----> {true}:
coset-polynomial: -----> {true}:
protocol-emulation: -----> {network}:
signal-type: -----> {loopstart}:
ds1-group-number: -----> {0}:
line-power: -----> {disabled}:
timeslot-assignment: ----->
{0+1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20+21+22+23}:
transmit-clock-adaptive-quality: --> {stratum3}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

- 3 In the **system-clock-profile**, enable the clock source and change the default weight (if necessary):

```
zSH> update system-clock-profile 1-4-1-0/ds1
system-clock-profile 1-4-1-0/ds1
Please provide the following: [q]uit.
system-clock-eligibility: -> {false}: true
system-clock-weight: -----> {5}:modify the weight if necessary
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Configure MXK line and uplink cards for system clocking

This section describes how to set the clock source from line and uplink cards and includes:

- [Set a line card as the clocking source, page 180](#)
- [Set a CLK or TOP uplink card as the clocking source, page 181](#)

Set a line card as the clocking source

Follow this procedure when the clocking source is taken from a line card.



Note: The TAC card type for Europe is *e1*.

Configuring a line card to be the primary synchronized network timing source

- 1 Enter **slots** to view cards available for synchronized network timing.

```
zSH> slots
MXK 819
```

Uplinks

a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)

b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)

Cards

```
1: MXK 20 ACT ETH (RUNNING)
10: MXK 8 PORT GPON (RUNNING)
11: MXK 4 PORT GPON (RUNNING)
14:*TAC ITM RING (RUNNING)
```

- 2 Change the **transmit-clock-source** parameter from *throughtiming* to *looptiming* on the **ds1-profile** of the available card.

```
zSH> update ds1-profile 1-14-1-0/ds1
ds1-profile 1-14-1-0/ds1
Please provide the following: [q]uit.
line-type: -----> {esf}:
line-code: -----> {b8zs}:
send-code: -----> {sendnocode}:
circuit-id: -----> {ds1}:
loopback-config: -----> {nolooop}:
signal-mode: -----> {none}:
fdl: -----> {fdlnone}:
dsx-line-length: -----> {dsx0}:
line-status_change-trap-enable: --> {enabled}:
channelization: -----> {disabled}:
ds1-mode: -----> {csu}:
```



```

csu-line-length: -----> {csu00}:
clock-source-eligible: -----> {eligible}:
transmit-clock-source: -----> {throughtiming}: looptiming
cell-scramble: -----> {true}:
coset-polynomial: -----> {true}:
protocol-emulation: -----> {network}:
signal-type: -----> {loopstart}:
dsl-group-number: -----> {0}:
line-power: -----> {disabled}:
timeslot-assignment: -----> {0}:
transmit-clock-recovery: -----> {none}:
transmit-clock-adaptive-quality: -> {stratum3}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

3 Configure the **system-clock-profile** of the available card and set the **system-clock-eligibility** parameter to *true*.

If necessary, set the **system-clock-weight** parameter with 10 as the most preferred and 1 is least preferred.

```

zSH> update system-clock-profile 1-14-1-0/ds1
system-clock-profile 1-14-1-0/ds1
Please provide the following: [q]uit.
system-clock-eligibility: -> {false}: true
system-clock-weight: -----> {5}:
.....u
Save changes? [s]ave, [c]hange or [q]uit: s

```

```

APR 16 14:00:43: warning: 1/a/1053: clkmgr: Secondary clock source set to 1/14/
1/0 Record updated.

```

```

zSH> APR 16 14:00:44: warning: 1/a/1053: clkmgr: System clock source set
to 1/14/1/0
APR 16 14:00:44: warning: 1/a/1053: clkmgr: There is no secondary clock

```

```

zSH> clkmgrshow
Primary system clock is 1/14/1/0 : T1
Secondary system clock is LOCAL timing

```

Set a CLK or TOP uplink card as the clocking source

In cases where the MXK is required for clocking, it is possible to use the CLOCK T1/E1 port on the MXK uplink cards for CLK or TOP. The MXK chassis with this uplink card can also use an appropriate line card as the clocking source.

Configure the 6X1GE uplink card for either T1/E1 or BITS

When BITS is the clocking source, inserting a Y cable or an individual cable with a BITS clock source causes the hardware and software to automatically

switch to BITS. See the [MXK Ethernet Uplink Cards on page 607](#) chapter for more information on both T1/E1 and BITS clocking on the uplink card.



Note: The card type for Europe is *ts1*.

Configuring a 6x1GE uplink card to be the synchronized Network Timing source

- 1 Enter **slots** to view available uplink cards.

```
zSH> slots
MXK 819
Uplinks
  a: MXK SIX GIGE (RUNNING+TRAFFIC)
  b: *MXK SIX GIGE (RUNNING)
Cards
  2: MXK 24 PORT VDSL2 POTS (RUNNING)
  3: MXK 20 ACT ETH (RUNNING)
  5: MXK 72 PORT POTS (RUNNING)
 13: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (RUNNING)
```

- 2 Change the **transmit-clock-source** parameter from *throughtiming* to *looptiming*.

```
zSH> update ds1-profile 1-a-1-0/ds1
ds1-profile 1-a-1-0/ds1
Please provide the following: [q]uit.
line-type: -----> {esf}:
line-code: -----> {b8zs}:
send-code: -----> {sendnocode}:
circuit-id: -----> {ds1}:
loopback-config: -----> {nolop}:
signal-mode: -----> {none}:
fdl: -----> {fdlnone}:
dsx-line-length: -----> {dsx0}:
line-status_change-trap-enable: ---> {enabled}:
channelization: -----> {disabled}:
ds1-mode: -----> {csu}:
csu-line-length: -----> {csu00}:
clock-source-eligible: -----> {eligible}:
transmit-clock-source: -----> {throughtiming}: looptiming
cell-scramble: -----> {true}:
coset-polynomial: -----> {true}:
protocol-emulation: -----> {network}:
signal-type: -----> {loopstart}:
ds1-group-number: -----> {0}:
line-power: -----> {disabled}:
timeslot-assignment: -----> {0}:
transmit-clock-adaptive-quality: --> {stratum3}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

3 Configure the **system-clock-profile** of the available card and set the **system-clock-eligibility** parameter to *true*.

If necessary, set the **system-clock-weight** parameter with *10* as the most preferred and *1* is least preferred. The default is *5*.

```
zSH> update system-clock-profile 1-a-1-0/ds1
system-clock-profile 1-a-1-0/ds1
Please provide the following: [q]uit.
system-clock-eligibility: -> {false}: true
system-clock-weight: -----> {5}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
```

```
APR 16 14:00:43: warning: 1/a/1053: clkmgr: Secondary clock source set to 1/
a/1/0 Record updated.
```

```
zSH> APR 16 14:00:44: warning: 1/a/1053: clkmgr: System clock source set
to 1/a/1/0
```

```
APR 16 14:00:44: warning: 1/a/1053: clkmgr: There is no secondary clock
```

Verify the changes and the clocking source.

```
zSH> get system-clock-profile 1-a-1-0/ds1
system-clock-profile 1-a-1-0/ds1
system-clock-eligibility: -> {true}
system-clock-weight: -----> {5}
```

```
zSH> clkmgrshow
Primary system clock is 1/a/1/0 : T1
Secondary system clock is LOCAL timing
```

Precision Time Protocol (PTP) and SyncE clock management on the MXK

The MXK supports two protocols for clocking sources from the network on the MXK-UPLINK-2X10G-8X1G-TOP uplink card:

- Precision Time Protocol (PTP) and Boundary Clock
See [Ordinary clock and boundary clock PTP configurations, page 184](#).
- SyncE
See [SyncE clock management, page 198](#).

Ordinary clock and boundary clock PTP configurations

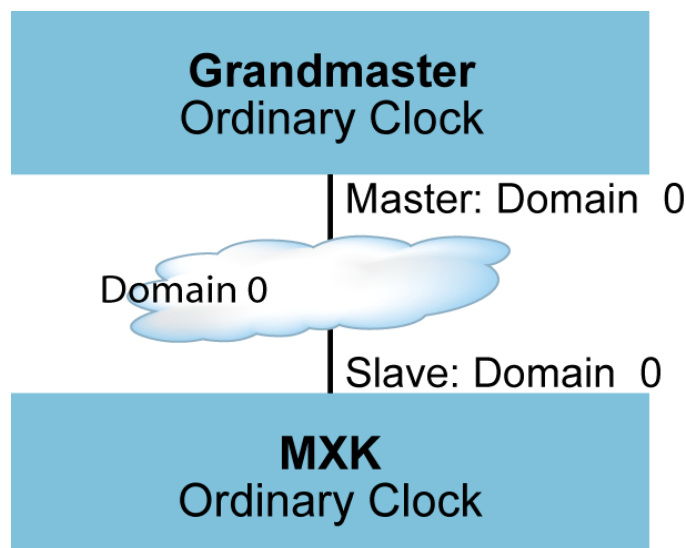
When Precision Time Protocol (PTP) is implemented on the MXK-UPLINK-2X10G-8X1GE-TOP, timing protocol message packets that measure timing are sent from a PTP Grand Master in the network to the MXK to provide accurate clocking to the TOP uplink card.

The MXK-UPLINK-2X10G-8X1GE-TOP supports two PTP clock modes -- Ordinary Clock and Boundary Clock -- as defined within IEEE 1588v2 (2008).

MXK Ordinary Clock

An MXK, configured as an Ordinary Clock, receives PTP timing protocol message packets from a Master Clock source in the network referred to as the Grand Master as shown in [Figure 10](#).

Figure 10: Ordinary clock in a one PTP configuration



Ordinary clock configurations support one PTP interface on one MXK. This PTP interface, configured as *slave*, communicates with the Grand Master and receives PTP timestamps on a single specified domain that matches the domain of the Grand Master as shown in [Figure 10](#).

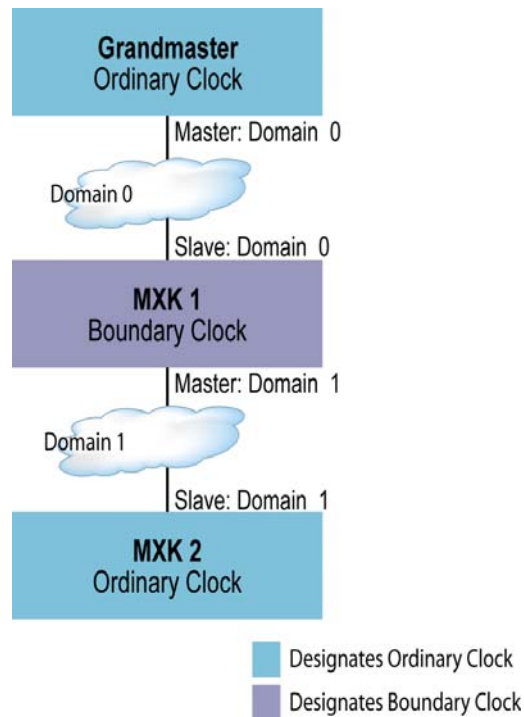
To implement Ordinary Clock:

- Must have a PTP Grand Master in the network to provide PTP packets. When primary and secondary Grand Masters are provisioned, the configuration is revertive.
- There is one PTP interface on a MXK.
- The MXK must have the MXK-UPLINK-2X10G-8X1G-TOP uplink card. PTP does not work on line cards.
- The domain of the PTP Grand Master(s) and the MXK must match and the MXK is configured in *slave* mode. See [Configuring PTP clock management for Ordinary Clock on page 186](#) for more information.

MXK Boundary Clock

The first MXK, configured as *boundary*, receives timing protocol messages from a Grand Master in the network on a single specified domain and sends timing protocol messages on a second specified domain to a second MXK configured as a *slave* as shown in [Figure 11](#).

Figure 11: Boundary clock configuration with multiple PTP interfaces



To implement Boundary Clock:

- Network segments are timing domains. There can be two timing domains, one domain for timing entering the boundary device from the PTP Grand Master, the second domain for the slave device receiving the timing information from the boundary device. See [Configuring PTP clock management for Boundary Clock on page 188](#) for more information.
- There are multiple PTP interfaces.
- When primary and secondary Grand Master clock sources are provisioned, the configuration is revertive and will return to the first device when it becomes again available.
- The MXK must have the MXK-UPLINK-2X10G-8X1G-TOP uplink card. PTP does not work on line cards.

This section also covers the following two procedures:

- [Configuring PTP clock management for Ordinary Clock, page 186](#)
- [Configuring PTP clock management for Boundary Clock, page 188](#)

Configuring PTP clock management for Ordinary Clock

When MXK is configured as an Ordinary Clock, it receives timing protocol message packets from a Grand Master clock source. There is a single PTP interface on the MXK that communicates with the Grand Master to maintain the timescale over a single specified time domain.

The PTP Grand Master in the network communicates with the TOP card on the MXK over IP and uses an *ipobridge* configured on the MXK.

To configure a MXK for Ordinary Clock management:

- 1 Create a bridge on a network facing Ethernet port with VLAN ID on the TOP uplink card.

See [Configure IP on a bridge for in-band device management overview on page 49](#) for more information on creating an IP on a bridge.

```
zSH> bridge add 1-a-5-0/eth tls vlan 3105 tagged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5/bridge
Bridge-path added successfully
```

- 2 Create an *ipobridge* for clock with the same VLAN ID.

```
zSH> interface add 1-a-6-0/ipobridge vlan 3105 10.51.5.5/24
Created ip-interface-record ipobridge-3105/ip.
```

Verify the bridge.

```
zSH> bridge show vlan 3105
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
tls              Tagged 3105  1/a/5/0/eth  ethernet5-3105/bridge  UP  D f8:66:f2:0d:3c:41
                                         D 68:ef:bd:c9:bc:65
                                         D 00:a0:12:19:43:a0
                                         D 00:01:47:b9:90:c7
```

```

ipobtlsl          Tagged 3105 1/a/6/0/ipobridge      ipobridge-3105/bridge      D 00:01:47:8b:d7:2d
                                                           UP S 00:01:47:18:07:43
                                                           S 10.51.5.5
2 Bridge Interfaces displayed

```

3 Create a route to the IP address.

```
zSH> route add default 10.51.5.254 1
```

Verify the route.

```

zSH> route show
Destination Routing Table
Dest                Nexthop                Cost    Owner                Fallback
-----
0.0.0.0/0           10.51.5.254           1       STATICLOW

```

4 Update the **ptp 1-a-1-0/ptp** profile with the information to connect the PTP Grand Master and the TOP uplink card.

You must provide the IP address of the PTP Grand Master that provides clock in the **acceptable-master-1** field and the *ipobridge* interface in the **ip-ifindex** field for clock to occur, the clock-mode is *slave*.

The domain **domain1MS** in the **ptp 1-a-1-0/ptp** profile must match the domain of the PTP Grand Master. The domain **domain2M** is not used.

```

zSH> update ptp 1-a-1-0/ptp
ptp 1-a-1-0/ptp
Please provide the following: [q]uit.
clock-mode: -----> {slave}: The mode of the MXK in relation to the PTP Grand Master is slave
sync-msg-interval: ---> {-5}:
announce-interval: ---> {1}:
delay-req-interval: --> {0}:
domain1MS: -----> {0}: domain must match the domain of the Grand Master
variance: -----> {32767}:
priority1: -----> {128}:
priority2: -----> {128}:
domain2M: -----> {0}: Domain remains unused in a Ordinary Clock configuration
ip-ifindex: -----> {0/0/0/0/0}: ipobridge-3105/ip
acceptable-master-1: -> {0.0.0.0}: 172.24.7.1 IP address of the PTP Grand Master
acceptable-master-2: -> {0.0.0.0}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Verify the changes.

```

zSH> get ptp 1-a-1-0/ptp
ptp 1-a-1-0/ptp
clock-mode: -----> {slave}
sync-msg-interval: ---> {-5}
announce-interval: ---> {1}
delay-req-interval: --> {0}
domain1MS: -----> {0}
variance: -----> {32767}

```

```

priority1: -----> {128}
priority2: -----> {128}
domain2M: -----> {0}
ip-ifindex: -----> {ipobridge-3105/ip}
acceptable-master-1: -> {172.24.7.1}
acceptable-master-2: -> {0.0.0.0}

```

- 5 Update the **1-a-1-0/ptp system-clock-profile** and set the system-clock-eligibility to *true* and the clock weight.

If there are multiple clock profiles with **system-clock-eligibility** set to *true*, the active clock with the highest weight will be selected as the system clock source.

See [system-clock-profile overview, page 177](#) for **system-clock-profile** configuration information.

```

zSH> update system-clock-profile 1-a-1-0/ptp
system-clock-profile 1-a-1-0/ptp
Please provide the following: [q]uit.
system-clock-eligibility: -> {false}: true
system-clock-weight: -----> {5}: 5
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Verify the changes.

```

zSH> get system-clock-profile 1-a-1-0/ptp
system-clock-profile 1-a-1-0/ptp
system-clock-eligibility: -> {true}
system-clock-weight: -----> {5}

```

Verify the clock source.

```

zSH> clkmgrshow
Primary system clock is 1/a/1/0 : PTP
BITS clock is not present
SyncE clock

```

Configuring PTP clock management for Boundary Clock

The PTP Grand Master in the network provides timing over packet to the MXKs configured as boundary clock devices. The **domain2M** parameter in the **ptp** profile of the devices configured for boundary clocking *must* match the domain of the PTP Grand Master in the network.

On the devices serving as boundary clock, clocking enters on the **domain2M** field and exits to the slave device on the **domain1MS** field.

The slave device receives clocking information on the **domain1MS** field from the boundary clock. The **domain1MS** value of the boundary clock(s) must match the **domain1MS** value found in the **ptp** profile of the slave device.

The PTP Grand Master in the network communicates with the client TOP cards on each MXK over IP, using *ipobridge* configured on the each of the MXKs.

1 Configure the first MXK for boundary clock.

- a** Create a bridge on a network facing Ethernet port with VLAN ID on the TOP uplink card on the first MXK device.

```
zSH> bridge add 1-a-2-0/eth t1s vlan 3410 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2/bridge
Bridge-path added successfully
```

See [Configure IP on a bridge for in-band device management overview on page 49](#) for more information on creating an IP on a bridge.

- b** Create an *ipobridge* for clocking with the VLAN ID and the desired IP address.

```
zSH> interface add 1-a-6-0/ipobridge vlan 3410 10.54.10.112/24
Created ip-interface-record ipobridge-3410/ip.
```

Verify the interface.

```
zSH> interface show
2 interfaces
```

Interface	Status	Rd/Address	Media/Dest Address	IfName
1/a/6/0/ip	UP	1 10.74.255.249/30	00:01:47:01:ad:b6	ipobridge-3461
1/a/6/0/ip	UP	1 10.54.10.112/24	00:01:47:01:ad:b6	ipobridge-3410

Verify the bridge.

```
zSH> bridge show vlan 3410
```

Type	Orig VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
t1s		Tagged 3410	1/a/2/0/eth	ethernet2-3410/bridge	UP	D f8:66:f2:0d:3c:41 D 54:75:d0:1b:a6:62 D 00:01:47:2b:b3:31
ipobt1s		Tagged 3410	1/a/6/0/ipobridge	ipobridge-3410/bridge	UP	S 00:01:47:01:ad:b6 S 10.54.10.112

2 Bridge Interfaces displayed

- c** Create a route to the IP address.

```
zSH> route add default 10.54.10.254/24 1
```

Verify the route.

```
zSH> route show
```

Destination	Routing Table	Nexthop	Cost	Owner	Fallback
Dest					

0.0.0.0/0

10.54.10.254

1

STATICLOW

- d** Update the **ptp 1-a-1-0/ptp** and the **ptp 1-b-1-0/ptp** profile with the information that connects the PTP Grand Master and the TOP uplink card of the first MXK acting as the boundary clock.

You must change the **clock-mode** to **boundary**, and provide the IP address of the PTP Grand Master that provides clock in the **acceptable-master-1** field and the **ipobridge** interface in the **ip-ifindex** field for clock to occur. You must also enter the **domain2M** and **domain1MS** values.

The **domain2M** domain must match the PTP Grand Master domain value and is where clock enters from the network. **domain1MS** is where clock is sent to the slave device and must match the **domain1MS** value of the **ptp** profile of the slave device.

```
zSH> update ptp 1-a-1-0/ptp
ptp 1-a-1-0/ptp
Please provide the following: [q]uit.
clock-mode: -----> {slave}: boundary
sync-msg-interval: ---> {-5}:
announce-interval: ---> {1}:
delay-req-interval: --> {0}:
domain1MS: -----> {0}: 1 domain must match the domain of the slave device
variance: -----> {32767}:
priority1: -----> {128}:
priority2: -----> {128}:
domain2M: -----> {0}: domain must match the domain of the PTP Grand Master
ip-ifindex: -----> {0/0/0/0/0}: ipobridge-3410/ip
acceptable-master-1: -> {0.0.0.0}: 172.24.7.1
acceptable-master-2: -> {0.0.0.0}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

```
zSH> update ptp 1-b-1-0/ptp
ptp 1-b-1-0/ptp
Please provide the following: [q]uit.
clock-mode: -----> {slave}: boundary
sync-msg-interval: ---> {-5}:
announce-interval: ---> {1}:
delay-req-interval: --> {0}:
domain1MS: -----> {0}: 1 domain must match the domain of the slave device
variance: -----> {32767}:
priority1: -----> {128}:
priority2: -----> {128}:
domain2M: -----> {0}: domain must match the domain of the PTP Grand Master
ip-ifindex: -----> {0/0/0/0/0}: ipobridge-3410/ip
acceptable-master-1: -> {0.0.0.0}: 172.24.7.1
acceptable-master-2: -> {0.0.0.0}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Verify the changes.

```
zSH> get ptp 1-a-1-0/ptp
ptp 1-a-1-0/ptp
clock-mode: -----> {boundary}
sync-msg-interval: ---> {-5}
announce-interval: ---> {1}
delay-req-interval: --> {0}
domain1MS: -----> {1}
variance: -----> {32767}
priority1: -----> {128}
priority2: -----> {128}
domain2M: -----> {0}
ip-ifindex: -----> {ipobridge-3410/ip}
acceptable-master-1: -> {172.24.7.1}
acceptable-master-2: -> {0.0.0.0}
```

- e** Update the **1-a-1-0/ptp system-clock-profile** and the **1-b-1-0/ptp system-clock-profile**, set the **system-clock-eligibility** to *true*, and enter the clock weight.

If there are multiple clock profiles with **system-clock-eligibility** set to *true*, the active clock with the highest weight will be selected as the system clock source.

See [system-clock-profile overview, page 177](#) for **system-clock-profile** configuration information.

```
zSH> update system-clock-profile 1-a-1-0/ptp
system-clock-profile 1-a-1-0/ptp
Please provide the following: [q]uit.
system-clock-eligibility: -> {false}: true
system-clock-weight: -----> {5}: 5
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

```
zSH> update system-clock-profile 1-b-1-0/ptp
system-clock-profile 1-b-1-0/ptp
Please provide the following: [q]uit.
system-clock-eligibility: -> {false}: true
system-clock-weight: -----> {5}: 5
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Verify the changes.

```
zSH> get system-clock-profile 1-a-1-0/ptp
system-clock-profile 1-a-1-0/ptp
system-clock-eligibility: -> {true}
system-clock-weight: -----> {5}
```

```
zSH> get system-clock-profile 1-b-1-0/ptp
system-clock-profile 1-b-1-0/ptp
```

```
system-clock-eligibility: -> {true}
system-clock-weight: -----> {5}
```

Verify the clock source.

```
zSH> clkmgrshow
Primary system clock is 1/a/1/0 : PTP
Secondary system clock is LOCAL timing
BITS clock is not present
```

2 Configure the second MXK for boundary clock.

- a** Create a bridge on a network facing Ethernet port with VLAN ID on the TOP uplink card on the first MXK device.

```
zSH> bridge add 1-a-4-0/eth tls vlan 3502 tagged
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-3502/bridge
Bridge-path added successfully
```

See [Configure IP on a bridge for in-band device management overview on page 49](#) for more information on creating an IP on a bridge.

- b** Create an *ipobridge* for clocking with the VLAN ID and desired IP address.

```
zSH> interface add 1-a-6-0/ipobridge vlan 3502 10.55.2.106/24
Created ip-interface-record ipobridge-3502/ip.
```

Verify the interface.

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 10.55.1.106/24    00:01:47:de:2e:70   ethernet1
1/a/6/0/ip     UP      1 10.55.2.106/24    00:01:47:8b:d7:30   ipobridge-3502
-----
```

Verify the bridge.

```
zSH> bridge show vlan 3502
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical          Bridge              St  Table Data
-----
ipobtls      Tagged  3502  1/a/6/0/ipobridge  ipobridge-3502/bridge  UP  S 00:01:47:8b:d7:30
S 10.55.2.106
tls          Tagged  3502  1/a/4/0/eth        ethernet4-3502/bridge  UP  D f8:66:f2:0d:3c:41
D 00:22:bd:c8:d2:50
D 00:01:47:a9:8a:c7
D 00:01:47:18:0c:12

2 Bridge Interfaces displayed
```

- c** Create a route to the IP address.

```
zSH> route add default 10.55.2.254/24 1
```

Verify the route.

```
zSH> route show
Destination Routing Table
Dest                Nexthop            Cost    Owner          Fallback
-----
0.0.0.0/0           10.55.2.254       1       STATICLOW
```

- d** Update the **ptp 1-a-1-0/ptp** and the **ptp 1-b-1-0/ptp** profile with the information that connects the PTP Grand Master and the TOP uplink card of the second MXK acting as the boundary clock.

You must change the **clock-mode** to **boundary**, and provide the IP address of the PTP Grand Master that provides clock in the **acceptable-master-1** field and the *ipobridge* interface in the **ip-ifindex** field for clock to occur. You must also enter the **domain2M** and **domain1MS** values.

The **domain2M** domain must match the PTP Grand Master domain value and is where clock enters from the network. **domain1MS** is where clock is sent to the slave device and must match the **domain1MS** value of the **ptp** profile of the slave device.

```
zSH> update ptp 1-a-1-0/ptp
ptp 1-a-1-0/ptp
Please provide the following: [q]uit.
clock-mode: -----> {slave}: boundary
sync-msg-interval: ---> {-5}:
announce-interval: ---> {1}:
delay-req-interval: --> {0}:
domain1MS: -----> {0}: 1 domain must match the domain of the slave device
variance: -----> {32767}:
priority1: -----> {128}:
priority2: -----> {128}:
domain2M: -----> {0}: domain must match the domain of the PTP Grand Master and the first MXK
ip-ifindex: -----> {0/0/0/0/0}: ipobridge-3502/ip
acceptable-master-1: -> {0.0.0.0}: 172.24.7.1
acceptable-master-2: -> {0.0.0.0}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

```
zSH> update ptp 1-b-1-0/ptp
ptp 1-b-1-0/ptp
Please provide the following: [q]uit.
clock-mode: -----> {slave}: boundary
sync-msg-interval: ---> {-5}:
announce-interval: ---> {1}:
delay-req-interval: --> {0}:
domain1MS: -----> {0}: 1 domain must match the domain of the slave device
variance: -----> {32767}:
priority1: -----> {128}:
priority2: -----> {128}:
domain2M: -----> {0}: domain must match the domain of the PTP Grand Master
```

```

ip-ifindex: -----> {0/0/0/0/0}: ipobridge-3502/ip
acceptable-master-1: -> {0.0.0.0}: 172.24.7.1
acceptable-master-2: -> {0.0.0.0}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Verify the changes.

```

zSH> get ptp 1-a-1-0/ptp
ptp 1-a-1-0/ptp
clock-mode: -----> {boundary}
sync-msg-interval: ---> {-5}
announce-interval: ---> {1}
delay-req-interval: --> {0}
domain1MS: -----> {1}
variance: -----> {32767}
priority1: -----> {128}
priority2: -----> {128}
domain2M: -----> {0}
ip-ifindex: -----> {ipobridge-3502/ip}
acceptable-master-1: -> {172.24.7.1}
acceptable-master-2: -> {0.0.0.0}

```

- e Update the **1-a-1-0/ptp system-clock-profile** and the **1-b-1-0/ptp system-clock-profile**, set the **system-clock-eligibility** to *true*, and enter the clock weight.

If there are multiple clock profiles with **system-clock-eligibility** set to *true*, the active clock with the highest weight will be selected as the system clock source.

See [system-clock-profile overview, page 177](#) for **system-clock-profile** configuration information.

```

zSH> update system-clock-profile 1-a-1-0/ptp
system-clock-profile 1-a-1-0/ptp
Please provide the following: [q]uit.
system-clock-eligibility: -> {false}: true
system-clock-weight: -----> {5}: 5
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

```

zSH> update system-clock-profile 1-b-1-0/ptp
system-clock-profile 1-b-1-0/ptp
Please provide the following: [q]uit.
system-clock-eligibility: -> {false}: true
system-clock-weight: -----> {5}: 5
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Verify the changes.

```

zSH> get system-clock-profile 1-a-1-0/ptp
system-clock-profile 1-a-1-0/ptp
system-clock-eligibility: -> {true}
system-clock-weight: -----> {5}

```

```

zSH> get system-clock-profile 1-b-1-0/ptp
system-clock-profile 1-b-1-0/ptp
system-clock-eligibility: -> {true}
system-clock-weight: -----> {5}

```

Verify the clock source.

```

zSH> clkmgrshow
Primary system clock is 1/a/1/0 : PTP
Secondary system clock is LOCAL timing
BITS clock is not present

```

3 Configure the slave device that will receive clock from the boundary clock.

- a Create a bridge on a network facing Ethernet port with VLAN ID on the TOP uplink card on the MXK device.

```

zSH> bridge add 1-a-6-0/eth tls vlan 3101 tagged
Adding bridge on 1-a-6-0/eth
Created bridge-interface-record ethernet6/bridge
Bridge-path added successfully

```

See [Configure IP on a bridge for in-band device management overview on page 49](#) for more information on creating an IP on a bridge.

- b Create an *ipobridge* for clocking with the VLAN ID and the desired IP address.

```

zSH> interface add 1-a-6-0/ipobridge vlan 3101 10.51.1.71/24
Created ip-interface-record ipobridge-3101/ip.

```

Verify the interface.

```

zSH> interface show
1 interface

```

Interface	Status	Rd/Address	Media/Dest Address	IfName
1/a/6/0/ip	UP	1 10.51.1.71/24	00:01:47:93:75:26	ipobridge-3101

Verify the bridge.

```

zSH> bridge show vlan 3101

```

Type	Orig VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
tls		Tagged 3101	1/a/4/0/eth	ethernet4-3101/bridge	UP	D f8:66:f2:0d:3c:41 D bc:ee:7b:e1:8c:b1 D 00:01:47:13:44:e6
ipobtls		Tagged 3101	1/a/6/0/ipobridge	ipobridge-3101/bridge	UP	S 00:01:47:93:75:26

2 Bridge Interfaces displayed

c Create a route to the IP address.

```
zSH> route add default 10.51.1.254/24 1
```

Verify the route.

```
zSH> route show
```

Destination Routing Table

Dest	Nexthop	Cost	Owner	Fallback
0.0.0.0/0	10.51.1.254	1	STATICLOW	

d Update the **ptp 1-a-1-0/ptp** profile and the **ptp 1-b-1-0/ptp** profile with the information that connects the masters and the slave for boundary clocking.

The **clock-mode** must be set to **slave** and the IP address of both the boundary clock PTP Grand Master are entered into the **acceptable-master-1** and **acceptable-master-2** fields. The **ipobridge** interface is entered in the **ip-ifindex** field for clock to occur. You must also enter the **domain2M** and **domain1MS** values.

The **domain2M** domain must match the PTP Grand Master domain value and is where clock enters from the network. **domain1MS** is where clock is received from the boundary device and must match the **domain1MS** value of the **ptp** profile of the boundary device.

```
zSH> update ptp 1-a-1-0/ptp
```

```
ptp 1-a-1-0/ptp
```

Please provide the following: [q]uit.

```
clock-mode: -----> {slave}: must be slave
```

```
sync-msg-interval: ---> {-5}:
```

```
announce-interval: ---> {1}:
```

```
delay-req-interval: --> {0}:
```

```
domain1MS: -----> {0}: 1
```

```
variance: -----> {32767}:
```

```
priority1: -----> {128}:
```

```
priority2: -----> {128}:
```

```
domain2M: -----> {0}: domain must match the domain of the PTP Grand Master
```

```
ip-ifindex: -----> {0/0/0/0/0}: ipobridge-3101/ip
```

```
acceptable-master-1: -> {0.0.0.0}: 10.54.10.112
```

```
acceptable-master-2: -> {0.0.0.0}: 10.55.2.106
```

```
.....
```

```
Save changes? [s]ave, [c]hange or [q]uit: s
```

```
Record updated.
```

```
zSH> update ptp 1-b-1-0/ptp
```

```
ptp 1-b-1-0/ptp
```

Please provide the following: [q]uit.

```
clock-mode: -----> {slave}: must be slave
```

```
sync-msg-interval: ---> {-5}:
```

```
announce-interval: ---> {1}:
```



```

delay-req-interval: --> {0}:
domain1MS: -----> {0}: 1
variance: -----> {32767}:
priority1: -----> {128}:
priority2: -----> {128}:
domain2M: -----> {0}: domain must match the domain of the PTP Grand Master
ip-ifindex: -----> {0/0/0/0/0}: ipobridge-3101/ip
acceptable-master-1: -> {0.0.0.0}: 10.54.10.112
acceptable-master-2: -> {0.0.0.0}:10.55.2.106
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Verify the changes.

```

zSH> get ptp 1-a-1-0/ptp
ptp 1-a-1-0/ptp
clock-mode: -----> {slave}
sync-msg-interval: ---> {-5}
announce-interval: ---> {1}
delay-req-interval: --> {0}
domain1MS: -----> {1}
variance: -----> {32767}
priority1: -----> {128}
priority2: -----> {128}
domain2M: -----> {0}
ip-ifindex: -----> {ipobridge-3101/ip}
acceptable-master-1: -> {10.54.10.112}
acceptable-master-2: -> {10.55.2.106}

```

- e** Update the **1-a-1-0/ptp system-clock-profile** and the **1-b-1-0/ptp system-clock-profile**, set the **system-clock-eligibility** to *true*, and enter the clock weight.

If there are multiple clock profiles with **system-clock-eligibility** set to *true*, the active clock with the highest weight will be selected as the system clock source.

See [system-clock-profile overview, page 177](#) for **system-clock-profile** configuration information.

```

zSH> update system-clock-profile 1-a-1-0/ptp
system-clock-profile 1-a-1-0/ptp
Please provide the following: [q]uit.
system-clock-eligibility: -> {false}: true
system-clock-weight: -----> {5}: 5
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

```

zSH> update system-clock-profile 1-b-1-0/ptp
system-clock-profile 1-b-1-0/ptp
Please provide the following: [q]uit.
system-clock-eligibility: -> {false}: true
system-clock-weight: -----> {5}: 5

```

```
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Verify the changes.

```
zSH> get system-clock-profile 1-a-1-0/ptp
system-clock-profile 1-a-1-0/ptp
system-clock-eligibility: -> {true}
system-clock-weight: -----> {5}
```

```
zSH> get system-clock-profile 1-b-1-0/ptp
system-clock-profile 1-b-1-0/ptp
system-clock-eligibility: -> {true}
system-clock-weight: -----> {5}
```

Verify the clock source.

```
zSH> clkmgrshow
Primary system clock is 1/a/1/0 : PTP
Secondary system clock is LOCAL timing
BITS clock is not present
```

SyncE clock management

Ethernet IP packet timing for port-to-port clock synchronization over the network.

Setting the system-clock-profile for SyncE

1 View current clock.

```
zSH> clkmgrshow
All lines are using LOCAL clock
```

2 Update the **system-clock-profile** by setting **system-clock-eligibility** to *true* on the designated Ethernet port for SyncE.

```
zSH> update system-clock-profile 1-a-2-0/eth
system-clock-profile 1-a-2-0/eth
Please provide the following: [q]uit.
system-clock-eligibility: -> {false}: true
system-clock-weight: -----> {5}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
FEB 10 10:02:39: warning: 1/a/1051: clkmgr: Secondary clock source set to 1/a/2/0/
eth Record updated.
zSH> FEB 10 10:02:40: warning: 1/a/1051: clkmgr: System clock
source set to 1/a/2/0/eth
FEB 10 10:02:40: warning: 1/a/1051: clkmgr: There is no secondary clock
```

Verify the clock source.

```
zSH> clkmgrshow
```

Primary system clock is 1/a/2/0 : ETHERNET Secondary system clock is LOCAL timing
BITS clock is not present

4

MXK BRIDGE CONFIGURATION

This chapter covers Zhone's bridging services including:

- [Overview of bridging on the MXK, page 201](#)
- [Basic bridged data on the MXK, page 239](#)
- [Advanced bridged data on the MXK with VLAN translation, page 298](#)
- [Filters for MXK bridges \(packet-rule-record\), page 313](#)
- [Additional bridging services, page 378](#)
- [MXK bridge statistics-on-demand, page 413](#)
- [Administrative commands, page 424](#)

Overview of bridging on the MXK

This section describes basic bridging topics:

- [Bridging fundamentals, page 201](#)
- [Terminology and concepts, page 203](#)
- [Tagging operations, page 209](#)
- [MXK bridge types, page 217](#)
- [bridge-path creation with the bridge add command, page 230](#)
- [IPv6 compatibility, page 235](#)

Bridging fundamentals

The main function of SLMS MSAPs and DSLAMs is to forward packets (IP routing) or frames (bridging).

Bridging services are primarily configured through the use of the **bridge add** command. The **bridge add** command creates a logical interface specifying the parameters for the bridge interface (bridge type, VLAN ID, tagging, COS options, and other parameters). This logical interface is stacked on a physical interface like an Ethernet, ADSL or GPON interface.

The bridging fundamentals described in this chapter do not intend to cover logical link layer bridging in an in depth or comprehensive manner, but are provided to highlight Zhone's mechanisms for providing bridging services.

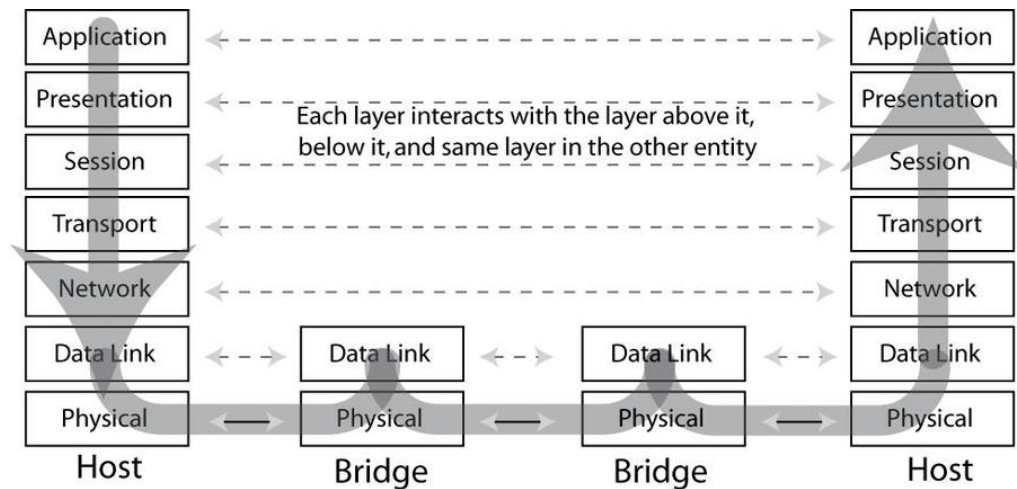
- Frames are delivered on MAC addresses (ISO Logical Link layer 2, bridging)
- Packets are delivered based on the IP address (ISO Network layer 3, routing)

The layers referred to above are part of the Open Systems Interconnection (OSI) reference model. While not all protocols follow the OSI model, the OSI model is helpful for understanding variations of network functionality.

Table 19: ISO Open Systems Interconnection Reference Mode I

Layer	Name	Function	
7.	Application	Network processes and application interactions	Host Layers
6.	Presentation	Mapping between application and lower layers — data presentation and encryption	
5.	Session	Manages connections between local and remote application.	
4.	Transport	Manages the end to end connection, reliability, tracks segments and retransmission (error control)	
3.	Network	Routing functions. Transferring data from source to destination. The best known layer 3 protocol is Internet Protocol (IP).	Media Layers
2.	Data Link	Transfers data between network entities.	
1.	Physical	Relationship between the transport medium (copper, fiber, wireless) and devices	

If an application on one host requests information from another networked application on another host (for example clicking a link to another page in a browser), the requests proceed down the layers until it is transmitted on the physical media (wire, fiber, wireless signal), until the message is picked up at the other end and progresses up the layers as shown in [Figure 12](#). The response follows the same process.

Figure 12: Applications requested networked information

Bridges direct frames based on address information in the frame as well as information learned from the processing and directing of other frames. The processing and directing of frames is the learning, forwarding, or filtering that is done by the device. The amount of processing and information read from the frame is kept to a minimum to enhance the throughput speed of the device.

Terminology and concepts

This section covers:

- [Physical port, page 204](#)
- [Physical interface, page 204](#)
- [Logical interface, page 205](#)
- [Bridges and bridge interfaces, page 205](#)
- [VLANs and SLANs, untagged, tagged and stagged, page 205](#)
- [Upstream and downstream, page 208](#)
- [Broadcast, multicast, and unicast, page 209](#)

Zhone SLMS bridging services draw from many specifications to provide a comprehensive suite of capabilities — IEEE 802.1-2004 (basic L2 bridging capabilities), IEEE 802.1W (Rapid Spanning Tree), DSL-Forum TR-101 and TR-156 (Ethernet backhaul for access devices, VLAN capabilities). Often there is not one specification to draw a set of terminology. Zhone uses a combination of terms from accepted standards, specifications, or Zhone's own terminology where no clear industry accepted term exists.

It is important to understand how the physical relates to the conceptual in building networks.

Physical port

The physical port is the physical connection on a device, essentially the layer 1 physical port. Examples of physical ports include

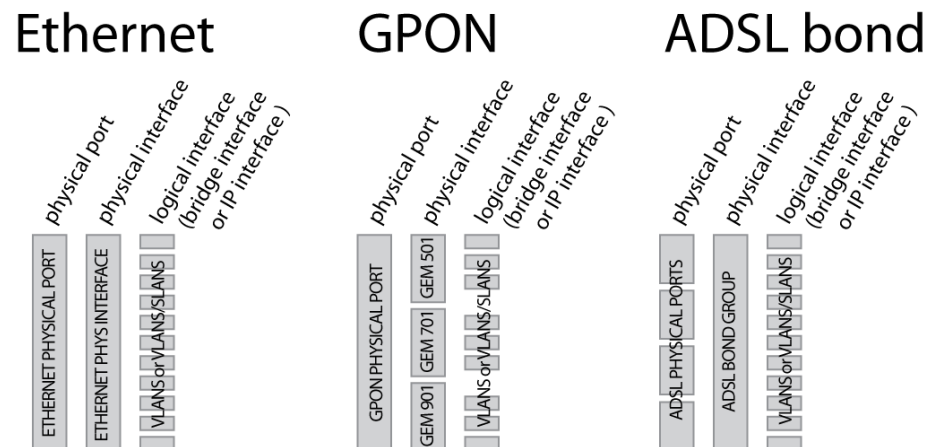
- Ethernet physical medium (Fast Ethernet or Gigabit Ethernet)
- Individual wire pair for POTs or xDSL
- GPON OLT port

The physical port is not necessarily the physical connector. A Champ connector may have 50 individual wire pairs. The physical port in this case, is the individual wire pair. The physical port in GPON would be one fiber connection, however that one connection may be and usually will be shared with multiple subscriber devices.

Physical interface

A physical interface is all of, a subset of, or a collection of, physical ports depending on the capabilities of the transportation technology as shown in [Figure 13](#).

Figure 13: Physical port to physical interface to logical interface vary by transport technology and bonding capabilities



The mapping of physical ports to physical interfaces may be:

- All of a physical port. With Ethernet, the physical interface is all of the physical port.
- A subset of a physical port. With GPON, GEM ports are used to separate a single physical port into multiple virtual ports.
- A collection of physical ports. Bonded links or IMA groups combine multiple physical ports into one logical interface.

Logical interfaces are associated with physical interfaces.

Logical interface

There are two types of logical interfaces — bridge interfaces and IP interfaces. These interfaces may be associated with all or part of the traffic on a physical interface. When the logical interface is broken down into connections, these connections are identified by a Virtual Local Area Network (VLAN) identifier, an ATM Virtual Connection (for connection based technologies such as ADSL), or both.

Bridges and bridge interfaces

A bridge is a collection of bridge interfaces which share a common attribute to form a community of interest. The attribute which defines the community of interest is either a VLAN ID or a combination of SLAN ID and VLAN ID.

Frames received on an interface belonging to a bridge can only be sent to other interfaces in the system belonging to the same bridge. Many bridges can exist in the system at the same time, each one defined by the VLAN ID or SLAN ID/VLAN ID.

Bridges connect network segments. The ends of the bridge are the bridge interfaces as defined in the **bridge-interface-record** profile.

Unlike a repeater which has two interfaces and takes in data on one interface and pushes it out the other (normally to strengthen the signal) or a hub which has more than two interfaces and takes in data on one interface and pushes it out on all the other data interfaces — bridges are more complex. Bridges analyze the incoming data frames to determine where to forward each frame. Where the data comes in (ingress) and where the data goes out (egress) on the device are determined by the bridge configuration. Zhone primarily uses two types of bridges — Transparent LAN Services (TLS) bridges (which are called symmetric because all the bridge interfaces have the same behavior) and asymmetric bridges which can be broken down into three different bridge interface types, each with its own behavior. See [MXK bridge types on page 217](#).

Frames which ingress on one bridge interface are not forwarded back out that same bridge interface.

VLANs and SLANs, untagged, tagged and stagged

VLANs and SLANs are used to separate traffic. VLANs and SLANs are typically used to separate services such as in triple play scenarios (voice, video, and data). Voice and video services are provided from servers on private networks. The messages from the voice and video servers are similar and have the same priority, only the content is different. Data services come from a gateway to the public Internet and the content is not as similar as the voice or video.

VLANs separate the traffic of all services, so the known traffic is separated from the unknown traffic. This separation also provides the means for

handling traffic differently through the use of Quality of Service (QoS) markings to prioritize voice and video traffic.

The separation of traffic allows for other mechanisms such as:

- conforming traffic to policies as with bandwidth limiting

For details see [Bandwidth limiting by port and service, single and dual rate limiting on page 334](#)

- providing port-to-port security of users sharing a VLAN as with Destination MAC swapping.

For details see [Destination MAC swapping on page 352](#)

- inserting identification information for DHCP servers

For details see [DHCP on bridge packet rules \(DHCP relay, and Forbid OUI\) on page 323](#)

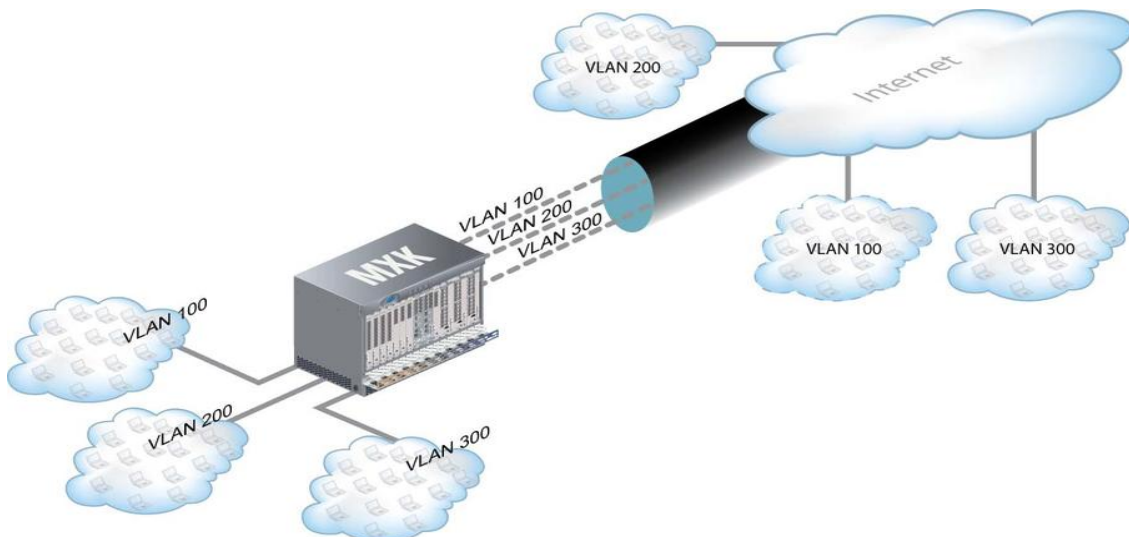
- inserting tags for identification purposes as when the MXK is a PPPoE intermediate agent

For details see [PPPoE with intermediate agent \(bridgeinsertpppoevendortag\) on page 327](#)

Another example of VLANs and SLANs is the separation of traffic for groups of hosts/users.

VLANs (and SLANs) may also be used for identifying the origination of frames as shown in [Figure 14](#). See [Tagging operations](#) for some network design scenarios.

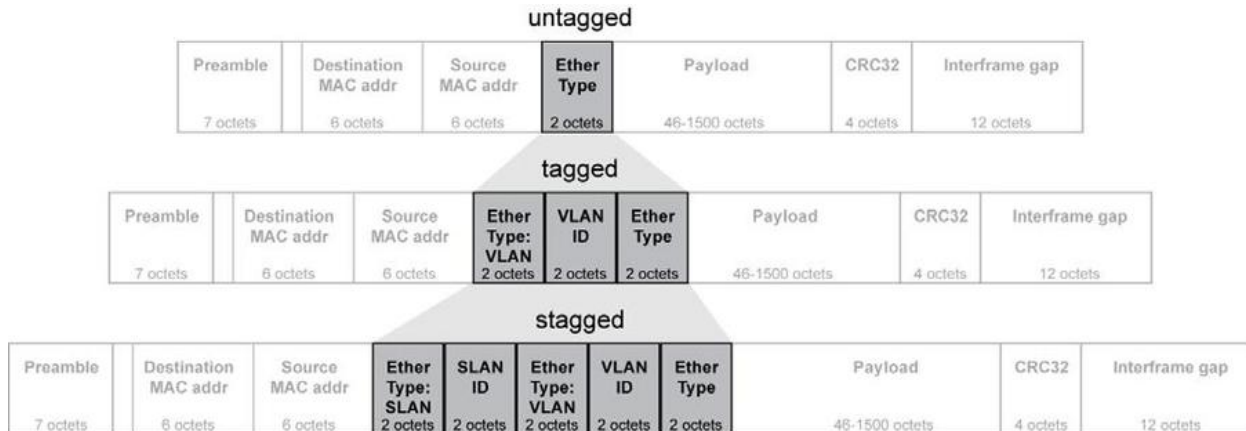
Figure 14: VLANs define to which bridge an incoming frame belongs



IEEE 802.1 Q-in-Q expanded the VLAN space in the Ethernet frame to support tagging already tagged frames. This second tag, an SLAN, creates a double-tagged Ethernet frame.

A frame which has no VLAN ID is referred to in the CLI as *untagged*. A frame which has a VLAN ID, but not an SLAN ID is single **tagged**, referred to as *tagged*. A frame which has both a VLAN ID and SLAN ID is double tagged, or *tagged* as shown in Figure 15.

Figure 15: Ethernet frames: untagged, single tagged and double tagged



Note: The octets for VLAN ID and SLAN ID include CoS information

Zhone's SLMS CLI uses a very flexible mechanism for defining bridge interfaces. When adding a bridge interface you can define the bridge interface to accept and send out *untagged*, *tagged* or *tagged* frames. No other frames will be accepted. If a bridge interface is expecting a tagged frame (using the **bridge add** command with the *tagged* key word to create the bridge interface), then untagged frames or double tagged frames will not be handled by this bridge interface. If a double tagged frame is expected, untagged and single tagged frames will not be handled by this interface. Those frames may be handled by other bridge interfaces depending on the configuration.

Only one untagged bridge interface can exist on a port or sub-port since frames will not have a VLAN ID to match multiple bridge interfaces. Untagged bridges are created using the **bridge add** command with either the *untagged* key word or not using the key words to define single tagged (*tagged*) or double tagged (*tagged*).

You can issue a **bridge add** command without specifying whether the bridge interface is to be *untagged*, *tagged* or *tagged*. However, Active Ethernet and VDSL downlinks are typically configured as *tagged* bridges even though the default is *untagged*. GPON downlinks *must* be configured *tagged*. ADSL is *untagged* as traffic is separated by vc. EFM SHDSL and T1/E1 downlinks can be either *tagged* or *untagged*.

If you do not designate a tagging option, the bridge interface assigns a default tagging based on the type of bridge interface.

- downlink, downlink-data, downlink-voice, downlink-pppoe, downlink-p2p, downlink-video

untagged

- uplink, intralink, downlink-upmcast (in this case, *tagged* must be designated with the **bridge add** command for the *downlink-upmcast* bridge-type is only on GPON downlinks)

tagged

- TLS

untagged

- wire

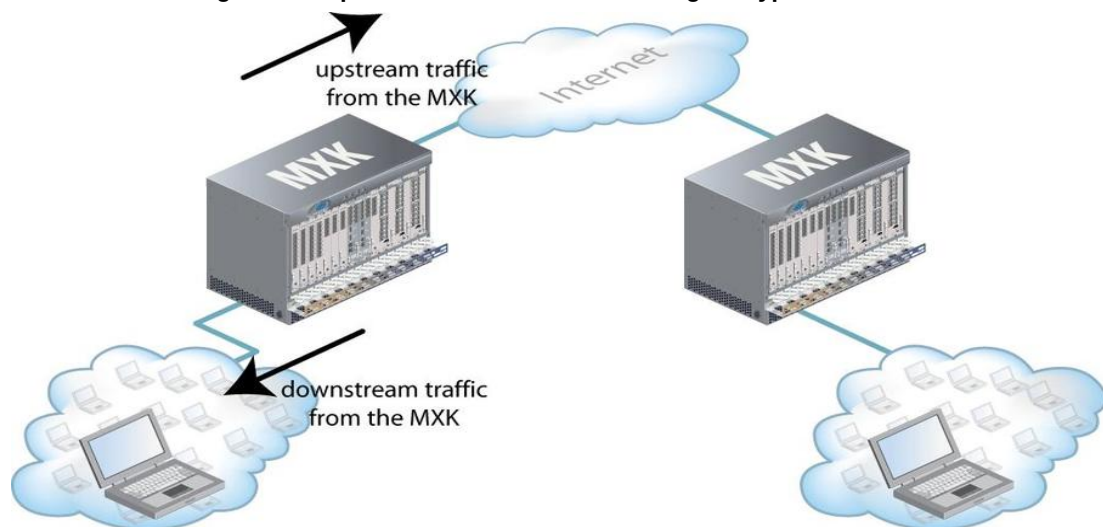
untagged Must designate a VLAN or SLAN.

See [Tagging operations on page 209](#) for more information on untagged, tagged, and staged traffic.

Upstream and downstream

Upstream and downstream are situational terms and are used in an SLMS device-centric manner. Typically the term upstream means the SLMS device's physical interface(s) are facing toward the core of the network and the term downstream means the device's physical interfaces is facing toward subscribers as described in [Figure 16](#).

Figure 16: Upstream and downstream using the typical model



This model assumes a hierarchy, but neglects the notion that at some point the data stream must change from upstream to downstream (since it is going from one application to another, one host to another, one user to another, even if one of the applications is a video server. To the server company, the data stream is going upstream to the core to get to the client). In other words, there is no way of defining “up” clearly throughout the entire conceptual model. Therefore the terms upstream and downstream are used with the general understanding that upstream is toward the Internet core and downstream is toward the subscriber.

The terms upstream and downstream are closely associated with the bridge interface types *uplink* and *downlink*. Uplinks and downlinks have different specific behaviors which define the bridges.

The terms upstream and downstream are also used when discussing TLS interfaces. TLS interfaces have the same behavior for both upstream and downstream interfaces which may be advantageous for certain access situations.

Broadcast, multicast, and unicast

The purpose of a bridge is to transmit frames. In general, frames are received on one interface and then are transmitted out on one or more other interfaces. There are three general ways to transmit frames:

- unicast

Unicast frames are sent to a specific address.

- multicast

Multicast frames are sent to a limited number of entities.

- broadcast

Broadcasts are sent to all available entities, usually all devices in a subnet as they can be a reasonably limited set of entities.

Learning on bridge interfaces means that the interface learns the source MAC address from the Ethernet frame of a received frame and the MAC address (as well as the VLAN and bridge interface upon which the MAC address was received) is put in the forwarding database. See source and destination addresses in [Figure 15](#) to see the structure of the Ethernet frame. When the learned MAC address from a previously received frame is the destination MAC address in an Ethernet frame the device forward the frame to the appropriate egress bridge interface.

There is no learning when receiving broadcast or multicast frames.

Each bridge type has a different behavior for learning the source address and forwarding to the destination of the received frame. The different behaviors in learning and forwarding are discussed in the following sections — TLS bridges and asymmetric bridges. The behavior of each bridge type with relation to the learning and forwarding behavior of unicast frames is also discussed in [MXK bridge types](#).

Tagging operations

This section describes VLAN and SLAN tagging operations including:

- [Tagging operations overview, page 210](#)
- [Common tagging operation scenarios, page 212](#)

Tagging operations overview

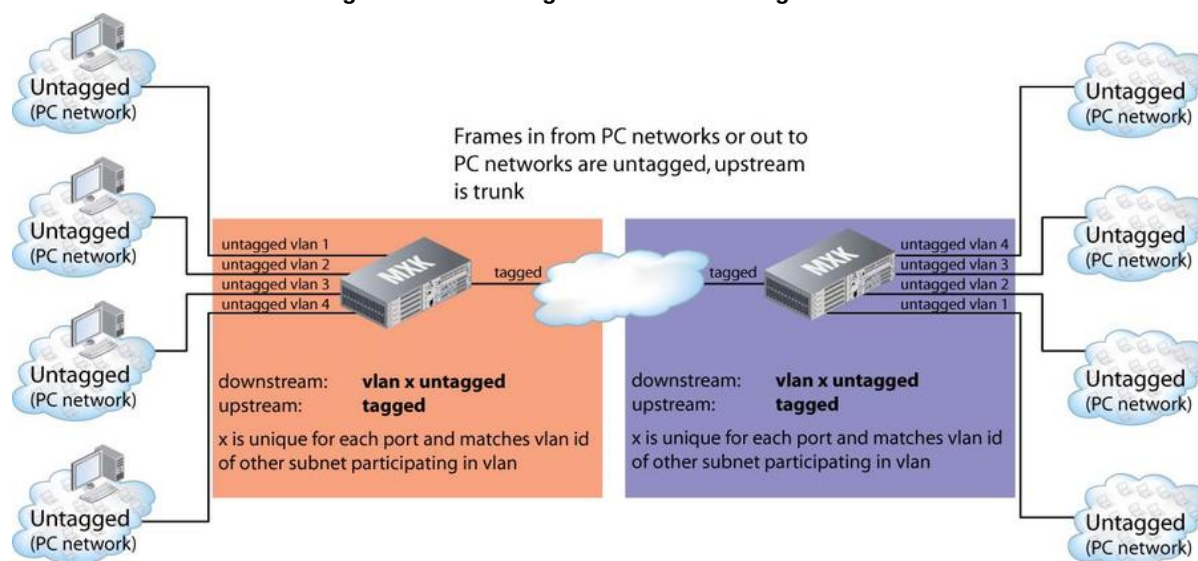
VLANs and SLANs (see [VLANs and SLANs, untagged, tagged and tagged](#), page 205 for information about VLANs and SLANs) define the bridge to which an incoming frame belongs. The bridge type — as discussed in [Section 4, MXK bridge types](#) — determines the forwarding behavior for the bridge. In conjunction with the forwarding and learning characteristics from the bridge types, you can also configure tagging operations.

Tagging operations provide the ability to configure interfaces for ingress filtering, VLAN/SLAN promotion, egress, and/or stripping.

Usually these tagging operations — ingress filtering, promotion, egress and/or stripping — are configured on downstream interfaces. Defining whether a bridge interface should be untagged, tagged or tagged depends on what the devices connected to the interface are expecting.

Zhone uses an extremely flexible mechanism for configuring tagging operations. Before discussing the various combinations which are possible, it is important to understand common cases, including the most common case — VLAN tagging for PC networks.

Figure 17: VLAN tags can be used to organize subnets



You can add a VLAN tag to all frames coming in from a PC network which has untagged Ethernet frames. However you want the PC network to be part of a virtual LAN with another remote PC network, so you configure the downstream bridge interface to accept the untagged frames and add a tag. Zhone uses the term promotion to signify adding the tag. The frames are then tagged frames and are sent out the upstream bridge interface tagged and directed to the remote PC network. The upstream bridge is a trunk line.

Likewise on receiving a frame from the remote PC network (which has the same VLAN tag), the frame is received on the uplink and forwarded to the proper downstream link because the VLAN ID matches (and assuming the destination MAC address of the unicast frame matches a learned MAC

address). However the PC network does not accept tags, so the VLAN tag is removed and the frame is forwarded to the device with the proper MAC address. Zhone uses the term stripping to signify removing VLAN and/or SLAN IDs.

In [Figure 17](#), the MXK is providing VLAN tags so on the other side of the cloud the frames may be forwarded to the proper VLANs as defined by the other MXK. In [Figure 17](#), the cloud may just be the cabling between two MXKs connected back to back; the cloud could also be a whole network of subtending MALCs, MXKs, the Internet, but the basic VLAN tagging is being done at the MXK devices at the network edge.

In the example from [Figure 17](#), the upstream interfaces are tagged with no VLAN ID designated. The downstream interfaces are untagged and given a VLAN ID which identifies which port (and hence which PC network) the frames received on these interfaces came from. This VLAN definition describes which VLAN tag to insert on ingress, and that VLAN ID upon receiving on the upstream interface on the remote MXK defines which downstream port to forward the frame. Since the downstream interface is untagged, the VLAN ID tag is stripped off and the frame sent out to the remote PC network.



Note: This example does not describe whether the bridges are asymmetric bridges or TLS bridges.

The four VLAN operations work together and are implied in the **bridge add (bridge modify)** command.

- Ingress filtering is the ability to have the bridge interface accept only frames with certain types of VLAN/SLAN tags.
- VLAN/SLAN promotion is the ability to add tags to a Ethernet frame. As with the example in [Figure 17](#), the VLAN tag defines membership in a VLAN (VLAN/SLAN defines membership with two tags).
- Egress is the reciprocal of ingress filtering and designates where to forward the frame based on VLAN, SLAN, or VLAN/SLAN tags. If a frame is received into the device and possibly promoted, then needs to find the other bridge interface(s) for egress.
- Stripping is the reverse of promotion. Stripping is removing the VLAN, SLAN or VLAN/SLAN tags.

Promotion and stripping always occur together. Filtering on ingress assumes the incoming frames already have at least one tag; you may filter on VLAN and also promote an SLAN. Receiving the internally forwarded frame to the egress assumes that the frame either has been received with tags or has been promoted to have tags.

See [Common tagging operation scenarios on page 212](#) using graphic representations to show the changes in frames as they are received on an interface forwarded to an egress interface and possibly promoted or stripped.

Zhone does not support tagged with known VLAN ID and unknown SLAN ID.



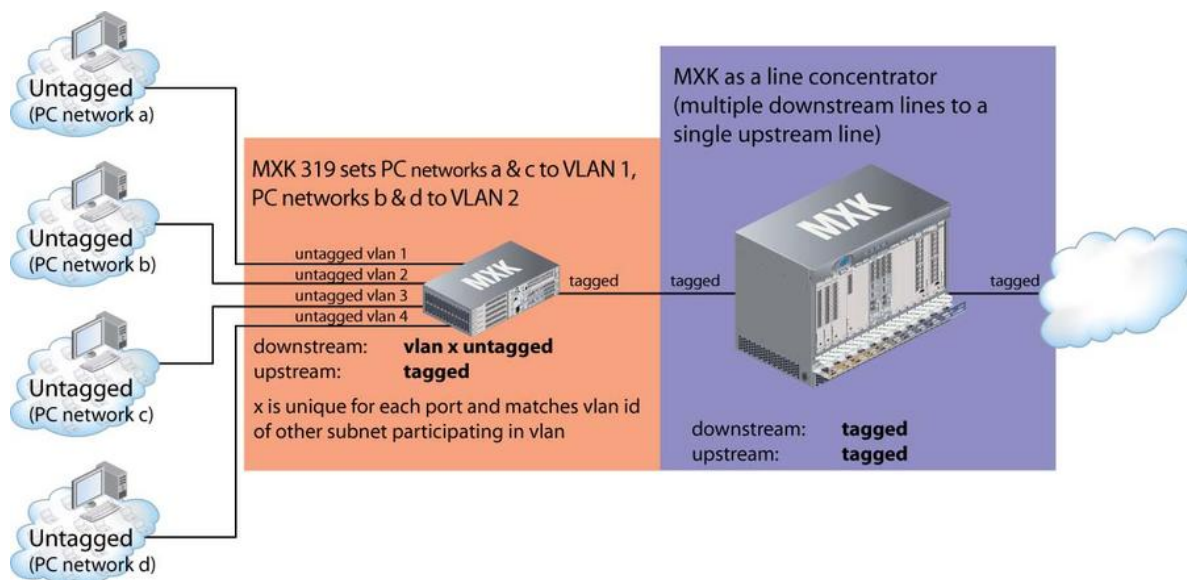
Note: The MXK does not support tagged frames with unknown VLAN and unknown SLAN.

The frames which come into the MXK are untagged, tagged and double tagged.

Common tagging operation scenarios

Figure 18 shows promoting untagged frames on the downstream interface (and so filtering to that interface when a frame with that VLAN ID is received on the upstream interface — given that the other bridging fundamentals are met, such as the MAC address as well as the VLAN ID match in the forwarding table if it is a downlink).

Figure 18: MXK 319 providing edge tagging, MXK as line concentrator



The untagged frame is accepted on the downstream interface, then it is promoted by inserting a VLAN ID. The upstream is tagged, so the tagged frame is sent out the upstream interface.

In order to complete the overlay with tagging and bridge types it helps to understand the following: the tagged frame will go out the uplink if part of an asymmetric bridge; if a TLS bridge, the frame will go where the forwarding table says it should go — the upstream interface if the MAC address matches. If the MAC address does not match addresses in the forwarding table the frame (an unknown unicast) would go out the upstream interface (along with the other participating bridge interfaces except the ingress bridge interface) since with TLS unknown unicasts are flooded out all member interfaces of the bridge.

A good way to learn tagging fundamentals is by exploring some of the common scenarios. Figure 17 shows promoting (and stripping) VLAN tags at the network edge. Figure 18 shows that same promotion at the edge, but now a line concentrator (in the example a MXK) distributes access from many downstream lines to a trunk. These multiple downstream subscriber lines could be from different transport technologies. In Figure 18 the MXK uses Ethernet frames. For the next example, Figure 20, the downstream devices could also be ADSL based.

ADSL technologies are based on ATM virtual connections. Another example of VLANs is terminating ATM from an xDSL modem and creating an Ethernet frame. In this case, the VLAN id is associated with the virtual channel. The ATM virtual connections can then be terminated and the data put into Ethernet frames with VLAN tags corresponding to the ATM virtual channel.

Figure 19: Parts of the bridge add command

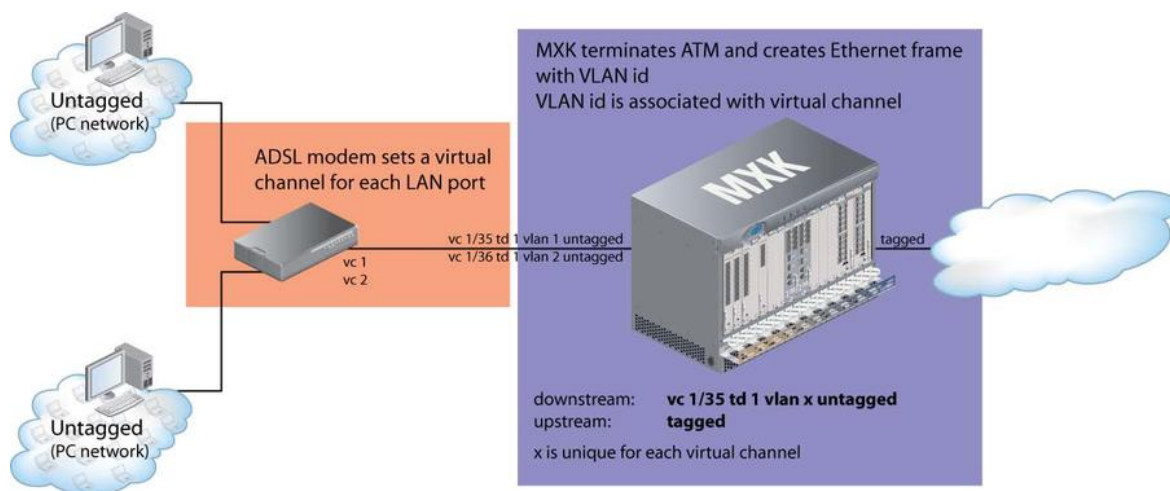
```
zSH> bridge add 1-7-1-0/shdsl vc 0/35 td 1 downlink tagged vlan 1
```

Diagram illustrating the parts of the `bridge add` command:

- `1-7-1-0/shdsl`: shelf/slot/port/sub-port
- `vc 0/35`: transport technology
- `td 1`: atm specific portion
- `downlink`: bridge interface type
- `tagged vlan 1`: VLAN and tagging

ADSL termination/Ethernet frame creation is a good example to show the parts of the **bridge add** command. Portions of the command define the bridging characteristics discussed in this chapter. The command also includes the transport technology and any associated information, such as the ATM specific portion for xDSL transport media.

Figure 20: ATM termination and Ethernet frame creation



Look at edge tagging in a tabular format to see that this same basic promotion concept works for different network.

The frame received on the downstream interface is untagged. Reading left to right, that frame is promoted to have a VLAN ID depending on the interface where the frame was received. The upstream interface is tagged, so a frame with a VLAN ID (but not double tagged) is forwarded to that interface. Since the bridge interface is tagged there is no stripping.

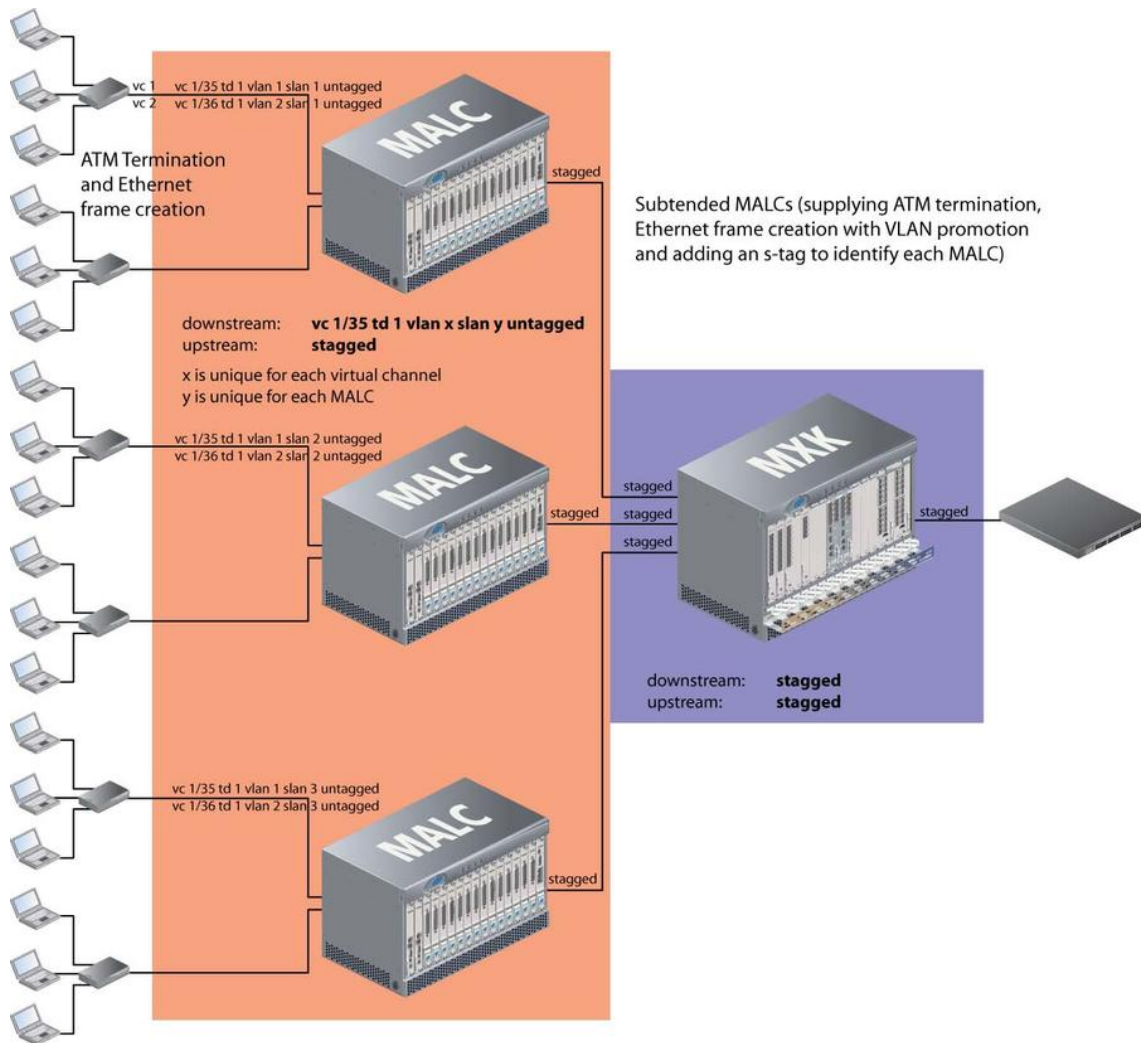
A frame on the upstream interface makes a reciprocal trip. A tagged frame is accepted on the upstream interface. Since no VLAN is defined it accepts all single tagged frames (so any VLAN ID). There is no promotion. The frame is forwarded to the bridge interface with the VLAN ID which matches the VLAN ID of the Ethernet frame. The egress interface is also untagged, so the VLAN ID is stripped out and the frame is sent to the network.

In this case multiple interfaces with the same VLAN are not being discussed, though that is a very common scenario. For the sake of discussion here, MAC addresses are found in the forwarding table for the egress interface.

All SLMS devices support tag promotion. How one defines the next level upstream from the edge of the network depends on the network architecture. In [Figure 21](#), the MALC is the next level up from the EtherXtend and acts as line concentrator and the MXK is upstream from the MALC. The example shows only VLAN tagging, but any of the SLMS devices could promote an s-tag, depending on what is necessary in the application or the overall network architecture.

[Figure 21](#) describes the next step upstream and describes double tags (the second tag are also called s-tags). In a subtended scenario you can add an s-tag for tracking the origination of the frame, perhaps by department. The example in [Figure 21](#) shows the double promotion of tags. The example shows the MALC providing ATM termination and the linkage to a VLAN ID and the promotion of an s-tag as well.

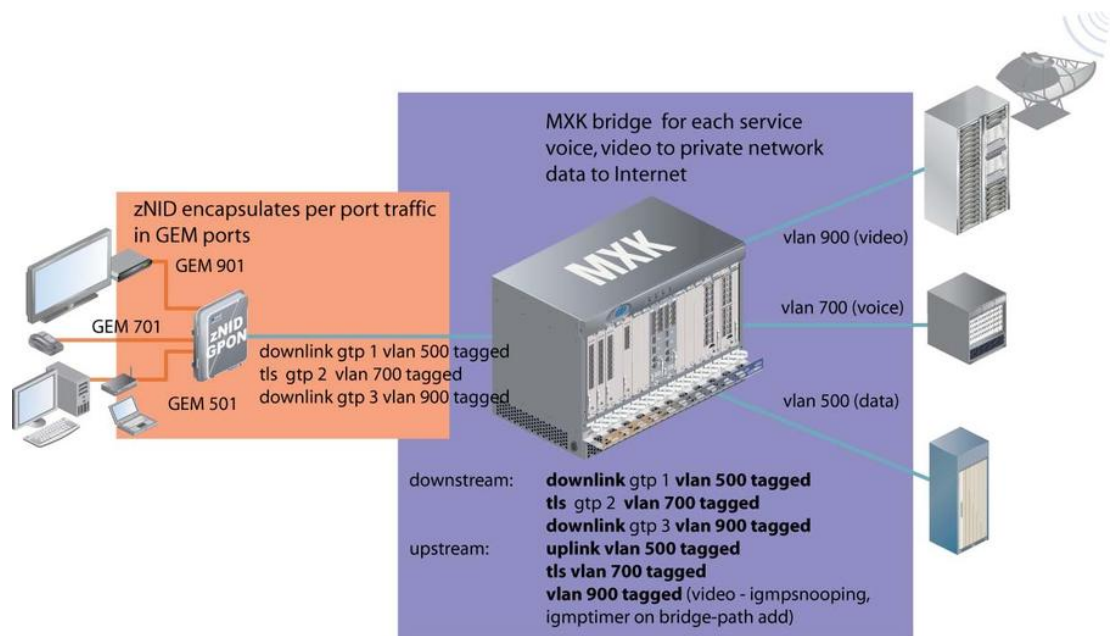
Figure 21: Q in Q supports adding a second tag



In Figure 21 describing the subtended MALCs, ingress frames received on the downstream bridge interface have both VLAN and SLAN IDs promoted. In this case the VLAN ID defines the ATM virtual channel. The SLAN ID designates from which MALC the frame originated.

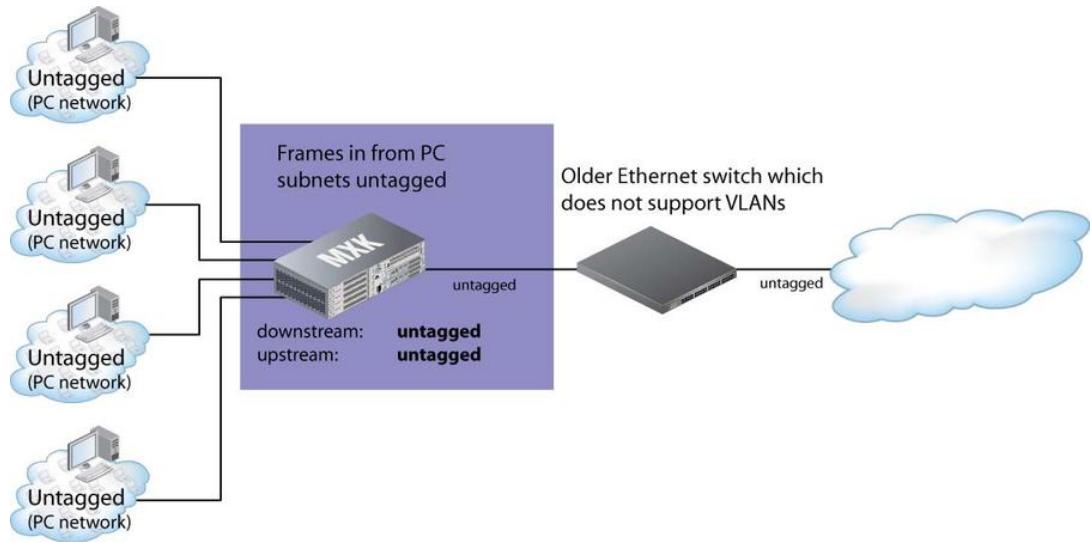
Uplinks are usually separated by VLAN IDs (see [VLANs and SLANs, untagged, tagged and staged](#)). Normally a triple play scenario separates traffic by VLAN ID for video, data, and voice services in order to configure QoS prioritization bridging filters.

Figure 22: OMCI GPON GEM port encapsulation to separate private VLANs



The flexibility of the SLMS tagging mechanism works for many scenarios. Not only do SLMS devices support many transport media technologies, but they also support all levels of tagging on both the downstream and upstream interfaces.

Figure 23: SLMS devices support untagged on upstream interface



To separate untagged information where there is other traffic on VLAN 0 (untagged frames which do not belong to a VLAN), you could tag on ingress and strip that tag on egress.

MXK bridge types

This section discusses bridge types on SLMS devices:

- [Symmetric bridges, page 217](#)
- [Asymmetric bridges, page 222](#)
- [Intralinked bridges, page 226](#)

The MXK uses two types of bridges — symmetric bridges which have the same bridging behavior and asymmetric bridge which have different bridging behavior. The bridge interfaces for symmetric bridges provide the same bridging behavior and bridge interfaces for asymmetric bridges provide different bridging behavior. Uplink and downlink bridge configurations are the most common asymmetric bridges but intralink bridges are also asymmetric bridges. The different behavior for these bridge types are useful in creating network bridges.

Symmetric bridges

This section discusses how to create symmetric bridges and includes:

- [Symmetric bridging overview, page 217](#)
- [Configure a TLS bridge, page 220](#)

Symmetric bridging overview

Symmetric bridges use TLS and wire bridge interfaces:

- TLS bridge interfaces have the same behavior regardless of which ports are being bridged.

A TLS bridge interface is created with a **bridge add** command and *tls* keyword.

TLS bridge interfaces only work in conjunction with other TLS bridge interfaces. The bridge path is automatically created with default static bridge parameters.

- Wire bridge interfaces are a special type of TLS bridge interfaces, and have the same behavior regardless of the ports being bridged.

A wire bridge interface is created with the **bridge add** command and *wire* keyword.

A wire bridge is only connected to another wire bridge in a two bridge interface configuration and reserves a VLAN ID for two ports for the entire system.



Note: When a VLAN ID is used for two *wire* bridges, that VLAN ID cannot be used anywhere else on the MXK system.

Transparent LAN services (TLS) bridges are used when traffic needs to flow freely among a community of users.

For example, a school district may use TLS bridges to extend a LAN to multiple campuses. The remote campuses will appear to be on the same LAN segment even though they are geographically separated.

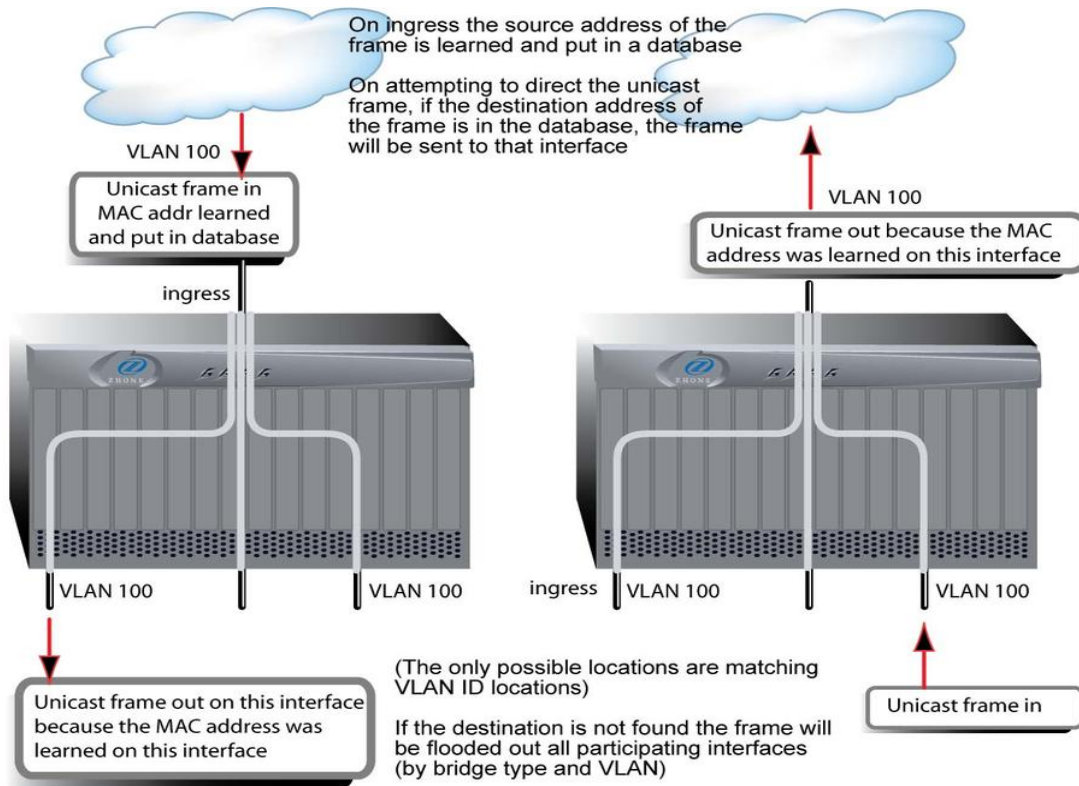
Another useful situation for TLS bridges are voice applications. The forwarding database does not retain information forever. Like all bridges, if there is no activity on the VoIP bridge, then the MAC address of the VoIP supplying access device will eventually time-out the MAC address of the VoIP in the bridge forwarding table. Upstream is the VoIP server which will try to send frames to the end VoIP supplying device. If no MAC address is in the forwarding table, the different type of bridges will behave differently. The TLS bridge will flood all the bridge interfaces of the TLS VoIP VLAN and rediscover the VoIP supplying access device. The downlink of an asymmetric bridge will, on the other hand, discard the frame, so the call will not be completed.

A TLS bridge interface is used only with other TLS bridge interfaces. TLS bridge interfaces are not used with any asymmetrical bridge interfaces.

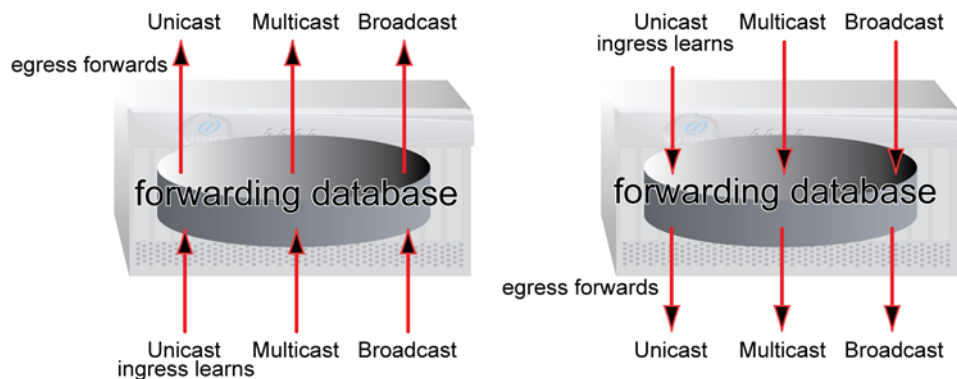
All interfaces in a TLS bridge are treated the same as shown in [Figure 24](#). There is no designation of an uplink or a downlink. When describing the equal interfaces of a TLS bridge it is helpful to think in terms of ingress or egress on an interface.

The default behavior of TLS bridges is to learn MAC addresses of unicast frames and forward the frames to learned destinations. TLS bridges do not flood IP TV multicast frames. Only unknown multicast and IPV4 reserved multicast frames are flooded.

Default wire bridge behavior is nonlearning with broadcasts and unicasts forwarded to all interfaces except the ingress interface.

Figure 24: In a TLS bridge all interfaces learn & forward the same

Frames entering the system on a TLS interface have their source MAC addresses learned and associated with the interface so that frames from the network that come in on other TLS bridges in the VLAN can be sent to the correct interface as shown in [Figure 25](#).

Figure 25: With TLS bridges all interfaces learn on ingress

Broadcasts and non-IGMP multicasts are flooded out all interfaces except for the interface where received

Source MAC addresses are learned on ingress of frames

Unicasts are forwarded to the known interface or if the interface is not in the forwarding database, they are flooded out all interfaces except for the interface where received.

Configure a TLS bridge

This example adds VLAN members to the VLAN 100 to create a community of traffic on the same VLAN.

For TLS bridges only, the first instance of a TLS bridge with VLAN ID, regardless of network facing or subscriber facing, associates a bridge-path with the configured VLAN ID.

For example, the first TLS bridge on a subscriber facing port for VLAN ID 444:

```
zSH> bridge add 1-6-4-0/eth tls vlan 444
```

```
Adding bridge on 1-6-4-0/eth
```

```
Created bridge-interface-record 1-6-4-0-eth/bridge
```

```
Bridge-path added successfully
```

```
zSH> bridge-path show
```

VLAN/SLAN	Bridge	Address
444	N/A	VLAN, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0, IGMP DSCP: 0, Flap Mode: Fast

The configurable parameters for the bridge-path that are relevant to TLS bridges are the aging period with a default of *3600*, and the flap control with a default of *fast*.

The default of *fast* indicates that as a MAC address comes into the system from one source and then is seen from another source, the MAC address table is purged from the first source and replaced with the second source without delay or restriction. If this behavior is not desired, the Flap Mode can be configured to *disabled* or *default*.

The default age of *3600* in seconds, is how long a MAC address is held in the MAC table before it is purged. This time is configurable on TLS bridges.

The MCAST and IGMP Query Interval are not relevant to TLS bridges.

Configuring a network facing TLS bridge and subscriber facing TLS bridges

TLS bridges can be thought of as a community since they share traffic much in the way a physical LAN shares traffic.

- 1 For each TLS bridge VLAN ID, configure one *tls* bridge interface on a network facing port.

```
zSH> bridge add 1-a-6-0/eth tls vlan 100
```

```
Adding bridge on 1-a-6-0/eth
```

```
Created bridge-interface-record ethernet6/bridge
```

```
Bridge-path added successfully
```

View the *tls* bridge:


```
SH> bridge show
```

Orig					
Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St Table Data
tls	100		1/a/6/0/eth	ethernet6/bridge	DWN

1 Bridge Interfaces displayed

View the TLS bridge-path.

```
zSH> bridge-path show
```

VLAN/SLAN	Bridge	Address
100 N/A		VLAN, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0, IGMP DSCP: 0, Flap Mode: Fast

- For each connection to the TLS bridge, VLAN ID, add a *tls* bridge interface to subscriber facing ports.

```
zSH> bridge add 1-6-1-0/eth tls vlan 100
```

Adding bridge on 1-6-1-0/eth

Created bridge-interface-record 1-6-1-0-eth/bridge

```
zSH> bridge add 1-6-2-0/eth tls vlan 100
```

Adding bridge on 1-6-2-0/eth

Created bridge-interface-record 1-6-2-0-eth/bridge

```
zSH> bridge add 1-6-3-0/eth tls vlan 100
```

Adding bridge on 1-6-3-0/eth

Created bridge-interface-record 1-6-3-0-eth/bridge

The TLS bridge interfaces with VLAN 100 will all work together as one TLS bridge.

- Use the **bridge show** command to view the *tls* bridges.

```
zSH> bridge show
```

Orig					
Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St Table Data
tls		100	1/6/1/0/eth	1-6-1-0-eth/bridge	UP
tls		100	1/6/2/0/eth	1-6-2-0-eth/bridge	DWN
tls		100	1/6/3/0/eth	1-6-3-0-eth/bridge	DWN
tls		100	1/a/6/0/eth	ethernet6/bridge	DWN

4 Bridge Interfaces displayed



Note: When you do not specify *untagged*, *tagged*, or *staged* to the bridge interface, the interface will use the default for TLS bridges, which is *untagged*.

Changing bridge-path defaults for TLS bridges

For TLS bridges, the bridge-path defaults are changed on the VLAN ID.

Change the parameters for an existing TLS bridge with VLAN ID of 100.

- View the existing bridge-path on VLAN ID.

```
zSH> bridge-path show
VLAN/SLAN   Bridge                               Address
-----
100 N/A                                           VLAN, Age: 3600, MCAST Age: 250, IGMP Query
Interval: 0, IGMP DSCP: 0, Flap Mode: Fast
```

2 Modify the existing bridge-path defaults.

```
zSH> bridge-path modify vlan 100 age 300 flap disable
Bridge-path /14/100/0/0/0/0/0/0/0 has been modified
```

3 View the changes.

```
zSH> bridge-path show
VLAN/SLAN   Bridge                               Address
-----
100 N/A                                           VLAN, Age: 300, MCAST Age: 250, IGMP
Query Interval: 0, IGMP DSCP: 0, Flap Mode: Disable
```

Asymmetric bridges

This section describes:

- [Asymmetric bridging overview, page 222](#)
- [Configure an uplink and downlink bridge, page 225](#)

Asymmetric bridging overview

Asymmetric bridges use three different bridge interface types:

- Uplinks

Uplinks are normally used for upstream traffic toward the Internet core.

An uplink bridge interface is created with a **bridge add** command and *uplink* keyword. The bridge path is automatically created with default static bridge parameters.

Uplink bridge interfaces only work in conjunction with asymmetric bridge interfaces.

- Downlinks

Downlinks are normally used for downstream traffic toward the subscribers.

A downlink bridge interface is created with a **bridge add** command and the *downlink*, *downlink-data*, *downlink-video*, *downlink-voice*, *downlink-upmcast*, *downlink-p2p*, or *downlink-pppoe* keywords.

Downlink bridge interfaces only work in conjunction with asymmetric bridge interfaces.

- Intralinks

Intralinks are normally used for subtending other SLMS devices.

An intralink bridge interface is created with a **bridge add** command and *intralink* keyword. The bridge path is automatically created.

Intralink bridge interfaces only work in conjunction with asymmetric bridge interfaces.

Asymmetric bridges are made up of one uplink and at least one downlink or intralink.

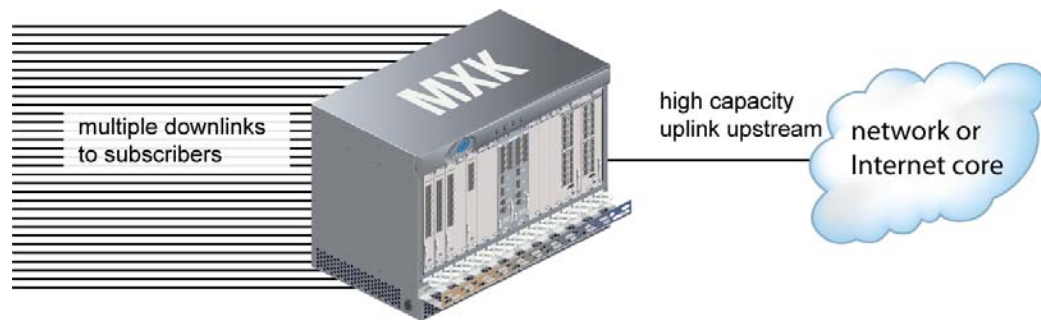
A single asymmetric bridge may use all three asymmetric bridge interface types — uplink, downlink, and intralink — however, a single bridge may only have one uplink. MXKs may have multiple intralinks per bridge, but other SLMS devices may only have one intralink. There may be multiple downlinks.

Most commonly there is one uplink and multiple downlinks as you would have with a line concentrator which splits a high capacity upstream link into multiple lower capacity downstream links.

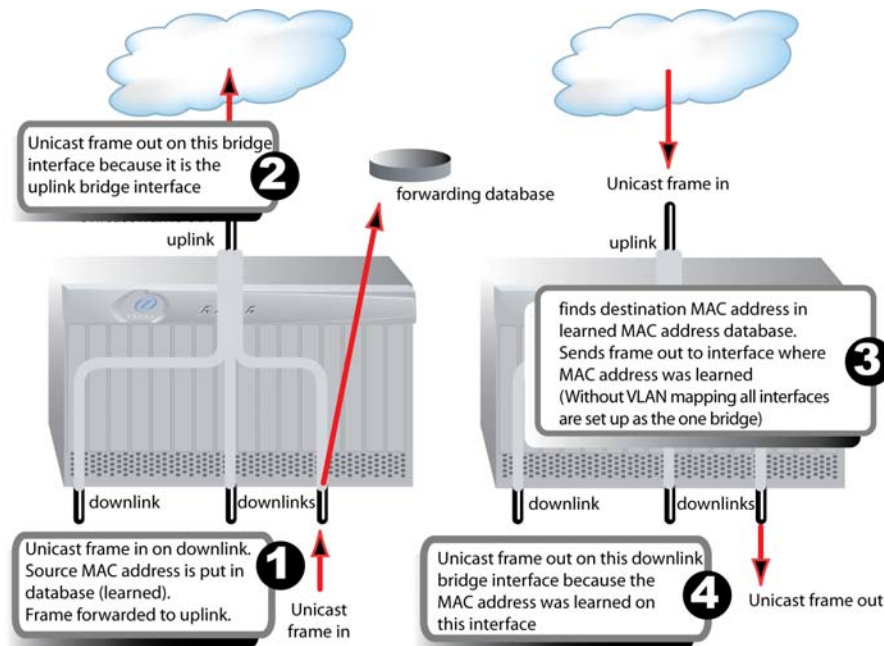
Intralink bridge interfaces are used for subtending other devices, usually other MXKs or MALCs. Intralinks have different learning behavior than uplinks or downlinks.

When setting up Internet access for multiple subscribers, configure the MXK as a line concentrator. The line concentrator model creates an asymmetric bridge with a high capacity link upstream configured as the uplink with many downlinks configured for subscribers.

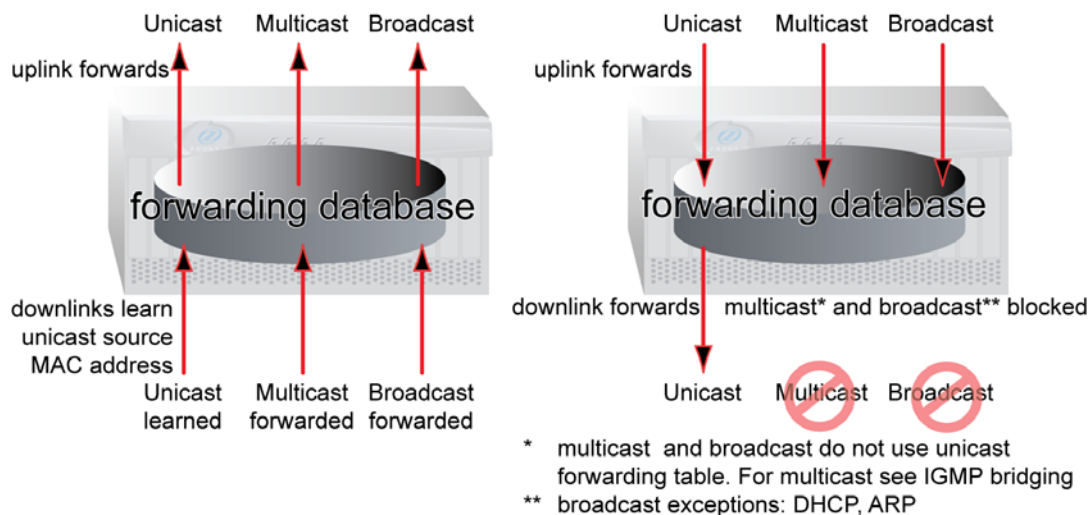
Figure 26: The line concentrator model



When a frame is received on a downlink bridge interface the source MAC address is learned and is put in the forwarding table along with the bridge interface and the VLAN on which the frame was received on. All frames whether unicast, multicast or broadcast received on downlinks are forwarded to the uplink as shown in [Figure 27](#).

Figure 27: Unicast forwarding and learning behavior for uplinks and downlinks

Unlike frames received on a downlink interface, when a unicast frame is received on an intralink bridge interface there is no learning and the frame is forced out the uplink as shown in [Figure 28](#).

Figure 28: Unicast forwarding and learning behavior for an asymmetric bridge

When frames ingress on an uplink the behavior of an asymmetric bridge is more complex.

When a unicast frame (a frame that is supposed to go to one address) is received on the uplink bridge interface and the address matches a learned MAC address, then the frame is forwarded to that address. Unknown unicast frames received on the uplink are discarded. (Unless there is an intralink, then unknown unicasts are sent on the intralink).

Broadcast frames have a special code in the address portion of the frame which identify it as a broadcast frame. These frames are normally duplicated and sent to all devices.

DHCP servers provide a pool of IP addresses, and upon request provide an IP address for a device. When a MXK acting as a DHCP relay agent receives a broadcast DHCP OFFER message on the uplink from a remote DHCP server the broadcast messages are forwarded to the MAC address if customDHCP is set to true. Otherwise, the broadcast DHCP messages are filtered.

Multicast is used when the same data is required by a group of clients at the same time. Unlike broadcast which sends to all devices, multicast provides content to a limited number of devices simultaneously. A common use of multicast would be video services. Receiving, duplicating and transmitting frames for high quality video to a large number of devices is processing time and capacity intensive. In multicast the number of recipients is guided by the multicast clients requesting to receive the multicast.

In an asymmetric bridge the general rule is that the source address of frames received on the downlinks are learned and the frames are sent out the uplink. Unicast frames received on the uplink are forwarded if found in the forwarding table, discarded if not. Multicasts and broadcasts received on the uplink are not forwarded with the DHCP and ARP exceptions noted above.

Configure an uplink and downlink bridge

All uplink bridges on the MXK require a VLAN ID. There must be an uplink bridge with a VLAN ID to match any existing downlink bridges with a VLAN ID in order to pass traffic. All uplink bridges default to *tagged* which means that the VLAN ID remains and is passed up to the network.

On the MXK, all bridge paths are set to *default*.

Configuring both the uplink and the intralink bridges with the **bridge add** command automatically creates the bridge path with the default configuration for that bridge. See [Bridge add and bridge-path modify defaults on page 232](#) for more information on when to use the **bridge-path modify** command when changing bridge-path defaults.

Configuring an uplink bridge and downlink bridges

- 1 Add a bridge interface on the uplink card.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 500 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-500/bridge
Bridge-path added successfully
```

The “1-a-2-0” string defines the shelf-slot-port-interface. This **bridge add** command defines the interface on port 2 of the card in slot *a*, as an Ethernet uplink bridge and the bridge-path is automatically created.

- 2 Add downlink bridge interfaces. In this case for data.

```
zSH> bridge add 1-1-6-0/eth downlink-data vlan 500 tagged
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth-500/bridge

zSH> bridge add 1-1-7-0/eth downlink-data vlan 500 tagged
Adding bridge on 1-1-7-0/eth
Created bridge-interface-record 1-1-7-0-eth-500/bridge
```

3 Verify the bridges.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
dwn-dat          Tagged 500    1/1/6/0/eth    1-1-6-0-eth-500/bridge    DWN
dwn-dat          Tagged 500    1/1/7/0/eth    1-1-7-0-eth-500/bridge    DWN
upl            Tagged 500    1/a/2/0/eth    ethernet2-500/bridge      UP  S VLAN 500 default
3 Bridge Interfaces displayed
```

Intralinked bridges

This section describes:

- [Intralinked bridging overview, page 226](#)
- [Configure intralinked MXKs, page 228](#)

Intralinked bridging overview

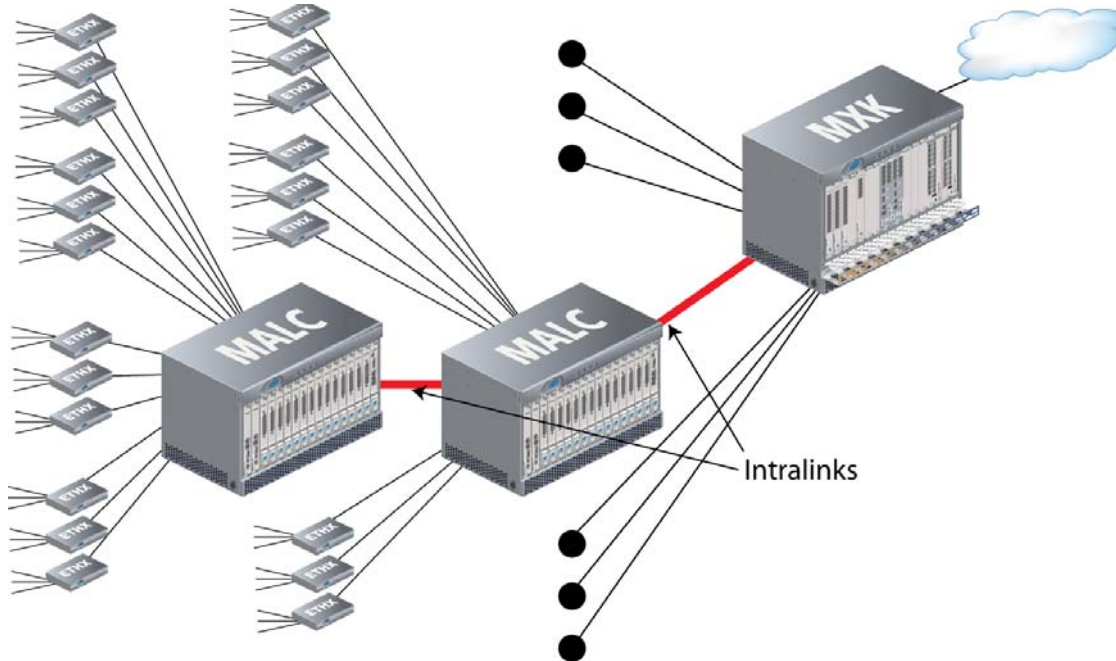
Intralinks basically daisy chain SLMS devices by sending all frames from the intralink interface to the uplink interface without learning.

The common case for an asymmetric bridge has the downlinks learning on sending and the uplinks forwarding on reception from outside of the MXK. If a frame is received on a downlink, the MAC address is learned. If the frame in on the uplink has a known address it is forwarded to the downlink that has that address. If the frame is unknown it is discarded.

In a case where you have multiple line concentrators linked, one below another, it is possible for the forwarding table on the head MXK in the chain or the upper MXKs to grow to an unmanageable size because they would be learning the MAC addresses of all devices downstream.

Intralink bridge interfaces, rather than learning the addresses connected to the intralink interface as they would from a downlink, merely send all frames from the intralink interface to the uplink without learning. The reciprocal behavior is that frames with unknown addresses received on the uplink interface are sent down the intralink interface.

[Figure 29](#) shows downlinks to EtherXtends and intralinks from an MXK to subtended MALCs. The intralink provides the means to forward all unknown frames received on the uplink to the intralink; the head device for the intralink does not need to learn the frames received on the intralink.

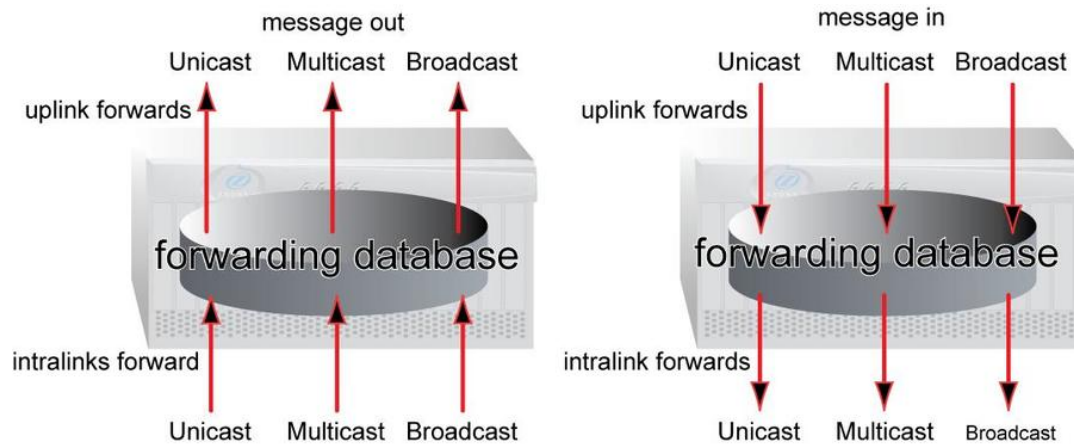
Figure 29: Line concentrator model with intralinks

An intralink bridge interface is used in conjunction with an uplink bridge interface, where the uplink bridge is the path upstream to the network. The intralink interface forwards traffic with unknown MAC addresses or multicasts to the uplink bridge-path without attempting to learn the addresses of the attached devices or network. Traffic coming into the intralink interface is forwarded to the uplink regardless of the destination MAC address. Broadcasts, multicasts, and unicasts (known and unknown) will be sent out the default interface, which is the uplink bridge for the VLAN.

In other words source addresses from an intralink interface are not learned, so the database of learned addresses will not add the address. Likewise when an unknown unicast frame is received on the uplink interface it will be transmitted to the intralink interface. Somewhere down the chain, the address may be known. Intralinks are normally used in conjunction with uplinks and can be used with downlinks.

Like uplinks, intralink bridge interfaces require the additional configuration of a bridge path. The bridge path sets a default intralink path for the specific VLAN onto the intralink bridge. If an intralink is missing the bridge path, traffic will not flow across the asymmetric VLAN.

Figure 30: The intralink portion of an asymmetric bridge



The general rule for intralinks is that input on the intralink is forwarded without the source address being learned. All frames with unknown addresses are forwarded to the intralink interface.

Configure intralinked MXKs

This example adds an intralink bridge interface to an asymmetric uplink/downlink bridge.

Configuring MXK intralinks

Configure the MXK for interlinked bridges.

- 1 Add bridge interfaces on the uplink card. For each VLAN ID designated on a downlink, there must be an uplink with the corresponding VLAN ID.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 101 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-101/bridge
Bridge-path added successfully
```

```
zSH> bridge add 1-a-2-0/eth uplink vlan 202 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-202/bridge
Bridge-path added successfully
```

- 2 Add downlink bridges for devices downstream from the MXK.

```
zSH> bridge add 1-1-6-0/eth downlink-data vlan 101 tagged
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth-101/bridge
```

```
zSH> bridge add 1-1-7-0/eth downlink-data vlan 202 tagged
Adding bridge on 1-1-7-0/eth
Created bridge-interface-record 1-1-7-0-eth-202/bridge
```

- 3 Create a bridge interface for the intralink with a VLAN ID.

The intralink can be between the MXK and a subtended MXK, MALC, or SLMS device. Then add the bridge path for the intralink.

```
zSH> bridge add 1-1-8-0/eth intralink vlan 444
Adding bridge on 1-1-8-0/eth
Created bridge-interface-record 1-1-8-0-eth-444/bridge
Bridge-path added successfully
```

This command mainly defines the behavior that source addresses from the intralink will not be learned.



Note: The MXK does not support the **global-intralink** keyword. For each VLAN or SLAN, you must define the bridge-path as an intralink using the **intralink** keyword.

This command defines the behavior that any frames with unknown addresses will be sent to the interlink with VLAN ID 444.

- 4 Create the uplink bridge for the intralink with the same VLAN ID for traffic to be passed to the network.

```
zSH> bridge add 1-a-3-0/eth uplink vlan 444 tagged
Adding bridge on 1-a-3-0/eth
Created bridge-interface-record ethernet3-444/bridge
Bridge-path added successfully
```

- 5 Verify the bridges created.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St Table Data
-----
dwn-dat          Tagged 101  1/1/6/0/eth  1-1-6-0-eth-101/bridge  DWN
dwn-dat          Tagged 202  1/1/7/0/eth  1-1-7-0-eth-202/bridge  DWN
int              Tagged 444  1/1/8/0/eth  1-1-8-0-eth-444/bridge  DWN S VLAN 444 Intralink
upl              Tagged 101  1/a/2/0/eth  ethernet2-101/bridge    UP  S VLAN 101 default
upl              Tagged 202  1/a/2/0/eth  ethernet2-202/bridge    UP  S VLAN 202 default
upl              Tagged 444  1/a/3/0/eth  ethernet3-444/bridge    DWN S VLAN 444 default
6 Bridge Interfaces displayed
```

- 6 Verify the bridge paths.

```
zSH> bridge-path show
VLAN/SLAN  Bridge      Address
-----
101 ethernet2-101/bridge  Default, Age: 3600, MCAST Age: 250, IGMP Query Interval:
0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym
202 ethernet2-202/bridge  Default, Age: 3600, MCAST Age: 250, IGMP Query Interval:
0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym
444 1-1-8-0-eth-444/bridge  Intralink
444 ethernet3-444/bridge  Default, Age: 3600, MCAST Age: 250, IGMP Query Interval:
0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym
```

bridge-path creation with the bridge add command

This section describes common bridging commands:

- [bridge add command, page 230](#)
- [bridge add parameters, page 230](#)
- [Verify the bridge-interface-record parameters, page 231](#)
- [Bridge add and bridge-path modify defaults, page 232](#)

bridge add command

The **bridge add** command combines the bridging type, the physical interface and the transportation media type, tagging operations, and other bridge rule additions such as packet rule records. (See [MXK bridge types on page 217](#), [Physical interface on page 204](#), [Tagging operations on page 209](#) and [Additional bridging services on page 378](#) for more detail).



Note: Entering general CLI commands on systems with large GPON configurations can take a long time to process. You must be as specific as possible with CLI commands. For example, **bridge flush all** should not be used. Instead, use commands based on the specific interface or MAC address.

bridge add parameters

The **bridge add** command designates the bridge interface type and the VLAN. The **bridge add** command also defines which card and port to add the bridge interface by the *shelf-slot-port-subport (or interface)/transport type* syntax. *shelf* is always 1. For example,

```
zSH> bridge add 1-a-2-0/eth uplink vlan 200
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-200/bridge
Bridge-path added successfully
```

adds a uplink bridge on the uplink card slot *a* port 2 with the VLAN ID 200.

For the MALC and the MXK, shelf is always 1 and slot is the physical slot where the card resides. For fixed units, like the MALC XP, Raptor XP and EtherXtend the shelf is always 1 and the slot is always 1. Port is the physical port. The subport be different depending on the *transport type*.

For GPON cards, the *transport type* is *gpon* and the subport is the GEM port. For Active Ethernet cards, the transport type is *eth* as in the example above and the subport is the logical interface. You may have multiple logical interfaces per port and the *subport* parameter identifies the logical interface.

Verify the bridge-interface-record parameters

When the **bridge add** command is entered, the system creates a **bridge-interface-record** profile. View the **bridge-interface-record** profile to verify the parameter settings, or when the **bridge-interface-record** profile defaults do not provide needed bridging behavior.

To verify the bridge interface settings, enter **get bridge-interface-record interface/type**.

```
zSH> get bridge-interface-record ethernet2-200/bridge
bridge-interface-record  ethernet2-200/bridge
vpi: -----> {0}
vci: -----> {0}
vlanId: -----> {200}
stripAndInsert: -----> {false}
customARP: -----> {true}
filterBroadcast: -----> {true}
learnIp: -----> {false}
learnUnicast: -----> {false}
maxUnicast: -----> {0}
learnMulticast: -----> {false}
forwardToUnicast: -----> {true}
forwardToMulticast: -----> {true}
forwardToDefault: -----> {false}
bridgeIfCustomDHCP: -----> {true}
bridgeIfIngressPacketRuleGroupIndex: -> {0}
vlanIdCOS: -----> {0}
outgoingCOSOption: -----> {disable}
outgoingCOSValue: -----> {0}
s-tagTPID: -----> {0x8100}
s-tagId: -----> {0}
s-tagStripAndInsert: -----> {true}
s-tagOutgoingCOSOption: -----> {s-tagdisable}
s-tagIdCOS: -----> {0}
s-tagOutgoingCOSValue: -----> {0}
mcastControlList: -----> {}
maxVideoStreams: -----> {0}
isPPPoA: -----> {false}
floodUnknown: -----> {false}
floodMulticast: -----> {false}
bridgeIfEgressPacketRuleGroupIndex: --> {0}
bridgeIfTableBasedFilter: -----> {NONE(0)}
bridgeIfDhcpLearn: -----> {NONE(0)}
mvrVlan: -----> {0}
vlan-xlate-from: -----> {0}
slan-xlate-from: -----> {0}
bridge-type: -----> {uplink}
```

The **bridge-interface-record** profile consists of a set of parameters. Designating a bridge type such as uplink, downlink, or TLS determines the parameter defaults that define the behavior of the bridge interface. Network facing and subscriber facing bridge interfaces work together to create the bridge.

Bridge add and bridge-path modify defaults

The system automatically creates a static bridge path with default values when entering the **bridge add** command for uplink, intralink, and TLS bridges. The default values are typically the type of bridge, uplink or intralink, the VLAN ID and the SLAN ID, and the tagged or untagged designation and configurable arguments such as age and IGMP query interval.

There are optional arguments for the bridge that must be configured from the CLI with the **bridge-path modify** command. These include:

- age
- multicast aging period (mcast)
- flap control (flap)
- unicast aging period (age)
- IGMP timer
- flags

When the **bridge-path modify** command is entered for an existing bridge, the previously existing bridge path is overwritten and unless otherwise specified, any previously existing optional arguments will revert to their default.

In other words, if the existing bridge path includes a designation for flap control and you want to add IGMP timer, you must enter both the flap control value *and* the IGMP timer value. Otherwise the flap control value will revert to the default.

For example, parameters such as *mcast* and *igmp* for video bridging, enter the **bridge-path modify** command with the proper variables.

The following example show a bridge added and the bridge-path automatically created.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 999 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-999/bridge
Bridge-path added successfully

zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN      Physical      Bridge      St  Table Data
-----
upl           Tagged 999  1/a/2/0/eth  ethernet2-999/bridge  DWN S VLAN 999 default
1 Bridge Interfaces displayed
```

Changing bridge-path defaults for uplink bridges

The following example shows the **bridge-path modify** command used to add a parameter that is not a default parameter, in this case to enable *igmpsnooping*, to the bridge path on the uplink bridge interface.

- 1 View the default bridge-path on the uplink bridge.

```
zSH> bridge-path show
```

VLAN/SLAN	Bridge	Address
999	ethernet2-999/bridge	Default, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym

- 2 Enter the **bridge-path modify** command to change the bridge path defaults.

The following example shows the **bridge-path modify** command used to add parameters to the bridge. In this case, the **igmpsendip enable** *<ipaddress>* is enabled to send a custom IP address.

```
zSH> bridge-path modify ethernet2-999/bridge vlan 999 default igmpsendip enable 172.16.1.3
Bridge-path ethernet2-999/bridge/3/999/0/0/0/0/0/0/0 has been modified
```



Note: For IPv6 compatibility use the **ipv6** keyword in the **bridge-path add/modify** command.

The default parameters are maintained and *igmpsnoothing* is *enabled*.

```
zSH> bridge-path show
```

VLAN/SLAN	Bridge	Address
999	ethernet2-999/bridge	Default, Age: 3600, MCAST Age: 241, IGMP Query Interval: 120, IGMP Proxy, Custom IP 172.16.1.3, IGMP DSCP: 0, Flap Mode: Default, Block: Asym

Changing bridge-path defaults for TLS bridges

The following example shows the **bridge-path modify** command used to change a parameter to the bridge path on the VLAN ID. For TLS bridges, the bridge-path exists on the VLAN ID.

Change the parameters for an existing TLS bridge with VLAN ID of 100.

- 1 Create a TLS bridge.

```
zSH> bridge add 1-a-2-0/eth tls vlan 100
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2/bridge
Bridge-path added successfully
```

- 2 View the bridge-path on the VLAN ID.

```
zSH> bridge-path show
```

VLAN/SLAN	Bridge	Address
100	N/A	VLAN, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0, IGMP DSCP: 0, Flap Mode: Fast

- 3 Modify the existing bridge-path defaults.

```
zSH> bridge-path modify vlan 100 age 300 flap disable
Bridge-path /14/100/0/0/0/0/0/0/0 has been modified
```

4 View the changes.

```
zSH> bridge-path show
VLAN/SLAN   Bridge                                     Address
-----
      100 N/A                                VLAN, Age: 300, MCAST Age: 250, IGMP Query
Interval: 0, IGMP DSCP: 0, Flap Mode: Disable
```

IPv6 compatibility

The MXK supports IPv6 for bridging. MXK Management interfaces and any other interfaces (such as PWE connections or POTS ports to softswitch) which require IP termination on the MXK use IPv4.

Bridging with IPv6 is quite similar to bridging with IPv4, however there are some differences. Whether the MXK IPv6 implementation uses Stateless and Stateful with IPv6 is determined by the downstream devices. Stateless uses Neighborhood Discovery Protocol (NDP). IPv6 and NDP and router advertisement are supported in all directions for TLS and Wire bridges. Asymmetric bridges support passing messages through in the downstream direction (from uplink to downlink/intralinks).

[Table 20](#) compares IPv4 and IPv6 configurations on the MXK and describes configuration differences for IPv6.

Table 20: IPv4 and IPv6 comparison

Feature/Configuration	Command	IPv4	IPv6		Comment
			Stateless	Statefull	
Management					
Management interfaces	interface add	Yes	No	No	Uses IPv4 only. This includes all IP termination on MXK in band and out of band IP addresses for management.
ipobridge (all types)	bridge add...vlan x interface add .../ ipobridge vlan x	Yes	No	No	Uses IPv4 only.
TLS/Wire bridges					
TLS bridges	bridge add ... tls	Yes	Yes	Yes	No config differences from IPv4, see TLS secure static bridge for IPv6 exception.
wire bridges	bridge add ... wire	Yes	Yes	Yes	No config differences from IPv4.
TLS Secure Static bridge	bridge-path add/ modify..	Yes	Yes	Yes	In bridge-path add/modify command use ipv6 keyword instead of ip.
Asymmetric bridges					
Uplink	bridge add ... uplink	Yes	Yes	Yes	No change.
Intralink	bridge add ... intralink	Yes	Yes	Yes	No change.
rlink	bridge add ... rlink	Yes	Yes	Yes	No change.
MVR	bridge add ... mvr	Yes	Yes	Yes	No change.

Table 20: IPv4 and IPv6 comparison

Feature/Configuration	Command	IPv4	IPv6		Comment
			Stateless	Statefull	
Video downlinks	bridge add ... downlink-video	Yes	No	No	Currently mutlicast video (IPTV) is only supported with IPv4 and IGMP. Not in IPv6 and MLD. Conceptually, there is no requirement for bridge type downlinkvideo to have certain IP version. There is no related provisioning.
Data downlinks	bridge add ... downlink-data	Yes	Yes	Yes	No video or voice on this downlink.
PPPoE downlinks	bridge add ... downlink-pppoe	Yes	Yes	Yes	Assumes no DHCP on this bridge.
P2P downlinks	bridge add ... downlink-p2p	Yes	Yes	Yes	No change.
Voice downlinks	bridge add ... downlink-voice	Yes	Supported for IP termination on downstream devices Not supported for IP termination on POTS ports which are on the MXK.		MXK is passing traffic (basic IPv6 support with asymmetric bridging). Not supported for IP termination (POTS ports to softswitch server). For voice on NIDs or CPEs voice traffic on downlinks is supported. From the MXK perspective where IP termination is down stream the configuration is just asymmetric bridging since there is no IP termination on the MXK.
Upstream multicast downlinks	bridge add ... downlink-upmcast	Yes			This scenario is a downlink where upstream multicast video streaming is allowed (for surveillance system, supported in GPON cards only).
Bridge features					
Secure DHCP	bridge add/modify ... secure	Yes	N/A. Uses NDP not DHCPv6	Yes. Autom atically creates bridge- paths for IPv6.	For IPv4: The secure option creates two static bridge paths (MAC and IP) for each host on the bridge that successfully negotiates its IP address from the DHCP server. For IPv6: Use bridge-path modify with ipv6 keyword.

Table 20: IPv4 and IPv6 comparison

Feature/Configuration	Command	IPv4	IPv6		Comment
			Stateless	Statefull	
Secure Static	bridge add secure static mac+ip	Yes	Yes. User would need to create two static bridge-paths. One for link-local and one for the global address.		For IPv4: use ip keyword and IPv4 IP format in bridge-path command For IPv6: use ipv6 keyword and IPv6 IP format in bridge-path command
DHCP relay	dhcp relay add ... and packet rule: rule add bridgeddhcprelay	Yes	No	No	Bridged dhcp relay is not supported in IPv6.
Option 82 insertion	packet rule: rule add bridgeinsertoption82	Yes	N/A. Uses NDP not DHCPv6	Yes	
Forbid OUI	packet rule: rule add bridgeforbidoui	Yes	Yes	Yes	
PPPoE with intermediate agent	packet rule: rule add bridgeinsertpppoeven dortag	Yes	Yes	Yes	
Rate limiting	packet rule: rule add ratelimitdiscard	Yes	Yes	Yes	
Color aware rate limiting	packet rule: rule add colorawareratelimitdiscard	Yes	Yes	Yes	
Bridge storm detection	packet rule: bridgestormdetect	Yes	Yes	Yes	
Destination MAC Swapping	packet rule: dstmacswapstatic	Yes	Yes	Yes	
Promote first encapsulations VLAN	packet rule: promotefirstencapsulationvlan	Yes	Yes	Yes	
Filter first encapsulation VLAN	packet rule: filterfirstencapsulationvlan	Yes	Yes	Yes	

Table 20: IPv4 and IPv6 comparison

Feature/Configuration	Command	IPv4	IPv6		Comment
			Stateless	Statefull	
Access Control List allow and deny	packet rule: rule add allow/deny	Yes	Yes	Yes	Access Control List has added several IPv6 options for rule add/deny: • ipv6 (v6 version of IP address) • icmp6 (IP proto 58) • srcipv6 (v6 version of srcip) • dstipv6 (v6 version of dstip) • dhcp6s (DHCPv6 server port 547) • dhcp6c (DHCPv6 client port 546)
Other connection types					
PWE connections	pwe-tdm add	Yes	No	No	Uses IPv4 only.
IGMP Send IP	bridge-path add/modify ... igmpsendip	Yes	No	No	Uses IPv4 only.
EAPS with voice	N/A	Yes	Not supported for IP termination on MXK POTS ports. Supported for IP termination on downstream devices.		Not supported for IP termination (POTS ports to softswitch server). Voice on NIDs or CPEs voice traffic on downlinks is supported. From the MXK perspective where IP termination is down stream the configuration is just asymmetric bridging since there is no IP termination on the MXK.
EAPS with PWE	N/A	Yes	Not supported for IP termination on MXK PWE ports. Supported for PWE IP termination on downstream devices.		PWE port to far end PWE port uses IPv4 only. Not supported for IPv6 termination (PWE port to far end PWE port). For PWE on NIDs PWE traffic on downlinks is supported. From the MXK perspective where IP termination is down stream the configuration is just asymmetric bridging since there is no IP termination on the MXK.

Basic bridged data on the MXK

This section includes the following bridging topics:

- [Uplink bridges with VLAN ID, page 239](#)
- [Downlink bridge-types for asymmetrical bridge configurations, page 240](#)
- [Downlink bridges with VLAN ID, page 243](#)
- [Q-in-Q on bridges \(VLAN IDs and SLAN IDs\), page 254](#)
- [Q-in-Q-in-Q \(VLAN IDs, SLAN IDs and packet rules\) on bridges, page 259](#)
- [Bridges using VLAN 0, page 263](#)
- [TLS bridges with VLAN ID, page 246](#)
- [Wire bridge configuration, page 250](#)
- [TLS bridge parameters floodUnknown and floodMulticast, page 247](#)
- [Bridges with link aggregation, page 271](#)
- [Bridge loop prevention, page 275](#)
- [Secure bridging, page 283](#)
- [Broadcast suppression, page 293](#)
- [Configure uplink and downlink bridges on GPON for triple-play services, page 294](#)

Uplink bridges with VLAN ID

All uplink bridges on the MXK require a VLAN ID. There must be an uplink bridge with a VLAN ID to match any existing downlink bridges with VLAN IDs in order to pass traffic. All uplink bridges default to *tagged* and the VLAN ID is passed up to the network.

On the MXK, all bridge paths are set to *default*.



Note: It is recommended not to change bridge default settings unless advanced bridge configuration is required.

See [Bridge add and bridge-path modify defaults on page 232](#) for when to accept the automatically created bridge path default configuration, and when it is necessary to enter the **bridge-path modify** command to create additional bridging configurations.

Creating an uplink bridge with VLAN ID

Create the uplink bridge.

- 1 Create the uplink bridge, then verify the bridge.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 300 tagged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-300/bridge
Bridge-path added successfully
```

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
upl           Tagged 300    1/a/5/0/eth      ethernet5-300/bridge  UP  S VLAN 300 default
1 Bridge Interfaces displayed
```

The *default* setting specifies this uplink receives all traffic with the designated VLAN ID from the downlinks.



Note: The MXK does not support the *global* variable.

2 Verify the **bridge-path** interface.

```
zSH> bridge-path show
VLAN/SLAN  Bridge          Address
-----
300 ethernet5-300/bridge  Default, Age: 3600, MCAST Age: 250, IGMP Query
Interval: 0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym
```

Deleting an uplink bridge

Delete the uplink bridge when necessary:

```
zSH> bridge delete ethernet5/bridge vlan 300
Bridge-path deleted successfully
ethernet5/bridge delete complete
```

Downlink bridge-types for asymmetrical bridge configurations

This section describes downlink bridge-types used for asymmetric bridge configuration depending on service provisioning requirements:

- [downlink-data bridging for data, page 241](#)
- [downlink-voice bridging for voice, page 241](#)
- [downlink-video bridging for video, page 241](#)
- [downlink-pppoe bridging for PPPoE, page 242](#)
- [downlink-p2p bridging for P2P, page 242](#)
- [downlink-upmcast bridging for upstream multicast, page 243](#)



Note: Depending on the service provisioned, downlink bridge-types are configured to provide the most efficient bridging behavior. Therefore, Zhone strongly encourages users to use the appropriate downlink bridge-types when creating asymmetrical bridge configurations.

downlink-data bridging for data

When service provisioning is for Internet access only, without video or voice, use the **bridge add** command with the *downlink-data* keyword. For example:

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 100
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth/bridge
```

Verify the bridge.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
dwn-dat          100    1/6/1/0/eth    1-6-1-0-eth/bridge    UP
1 Bridge Interfaces displayed
```

downlink-voice bridging for voice

When service provisioning for voice, use the **bridge add** command with the *downlink-voice* keyword. For example:

```
zSH> bridge add 1-1-6-0/eth downlink-voice vlan 200
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth/bridge
```

Verify the bridge.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
dwn-voi          200    1/1/6/0/eth    1-1-6-0-eth/bridge    UP
1 Bridge Interfaces displayed
```

downlink-video bridging for video

When service provisioning for video and the maximum number of video streams are greater than 0, use the **bridge add** command with the *downlink-video* keyword.

The downlink bridge is configured for video by entering the keyword **video** and the multicast control list and maximum number of video streams in the *m/n* format.

See [MXK basic bridged video configuration on page 429](#) for more information.

For example,

```
zSH> bridge add 1-1-6-0/eth downlink-video video 0/3
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth/bridge
```

Verify the bridge.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
dwn-vid           Untagged   1/1/6/0/eth   1-1-6-0-eth/bridge  UP
1 Bridge Interfaces displayed
```

downlink-pppoe bridging for PPPoE

When provisioning for data using PPPoE (without DHCP), use the **bridge add** command with the *downlink-pppoe* keyword.

For example,

```
zSH> bridge add 1-1-6-0/eth downlink-pppoe vlan 900
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth/bridge
```

Verify the bridge.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
dwn-poe           900        1/1/6/0/eth   1-1-6-0-eth/bridge  DWN
1 Bridge Interfaces displayed
```

downlink-p2p bridging for P2P

When provisioning a downlink for peer-to-peer, where users can view each others broadcast traffic and send unicast traffic directly within the MXK, use the **bridge add** command with the *downlink-p2p* keyword.

For example,

```
zSH> bridge add 1-1-6-0/eth downlink-p2p vlan 720
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth/bridge
```

Verify the bridge.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
dwn-p2p           720        1/1/6/0/eth   1-1-6-0-eth/bridge  DWN
```

1 Bridge Interfaces displayed

downlink-upmcast bridging for upstream multicast

When provisioning a downlink where upstream multicast video streaming is permitted, use the **bridge add** command with the *downlink-upmcast* and *tagged* keywords. The downlinks for this bridge type must be *tagged*, *untagged* bridges are not allowed.



Note: This bridge type is only supported on GPON cards.

For example,

```
zSH> bridge add 1-6-1-501/gponport gtp 1 downlink-upmcast vlan 100 tagged
Adding bridge on 1-6-1-501/gponport
Created bridge-interface-record 1-6-1-501-gponport-100/bridge
```

Verify the bridge.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
dwn-upm      Tagged 100  1/6/1/1/gpononu  1-6-1-501-gponport-100/bridge  DWN
1 Bridge Interfaces displayed
```

user specified bridging

User specific bridging.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
usr      Tagged 100  1/6/1/1/gpononu  1-6-1-501-gponport-100/bridge  DWN
1 Bridge Interfaces displayed
```

Downlink bridges with VLAN ID

This section discusses downlink bridge configurations:

- [Untagged downlink bridges on Active Ethernet, page 244](#)
- [Tagged downlink bridges on Active Ethernet, page 245](#)

You can configure downlink bridges on the MXK using the variables *tagged* or *untagged* depending on the downstream configuration and the downstream bridging behavior that you desire. See [Configuring an Active Ethernet untagged downlink VLAN bridge on page 244](#) and [Configuring an Active Ethernet tagged downlink VLAN bridge on page 245](#) for configuration procedures.



Note: It is recommended not to change the default settings unless advanced bridge configuration is required.

Untagged downlink bridges on Active Ethernet

Typically downlink bridges are *untagged* as many downstream devices do not expect or recognize VLAN IDs.

Specifying the downlink bridge as *untagged* causes the VLAN ID to be stripped out of the Ethernet packet on the way to the downstream device because it is not needed by the downstream device. When a packet is sent back toward the upstream connection, that VLAN ID is inserted back into the Ethernet packet.

If the correct VLAN ID is not on the packet traveling in the downstream direction, the packet will be dropped and not sent on to the downstream device. If that correct VLAN ID is not inserted back into the Ethernet packet traveling in the upstream direction, the uplink drops the packet.

The default for downlink bridges is *untagged*. Not designating either *untagged* or *tagged* when entering **bridge add interface/type downlink** always defaults to *untagged*. For example, both of these entries exhibit exactly the same bridging behavior.

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 200
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth/bridge
```

and

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 200 untagged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth/bridge
```

In some cases, downstream devices expect the VLAN ID. Entering **bridge add interface/type downlink tagged** causes the VLAN ID to remain in the Ethernet packet. In this case both upstream and downstream devices will recognize and accept the Ethernet packet.

Configuring an Active Ethernet untagged downlink VLAN bridge

Untagged downlink bridges are usually configured on Active Ethernet. To configure an untagged downlink bridge with a VLAN ID:

- 1 To create an *untagged* bridge for downstream connections enter **bridge add interface/type downlink vlan <id>**.

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 300 untagged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth/bridge
```


This example creates an untagged downlink interface on the Active Ethernet port 1 with a VLAN ID of 300.

- 2 To verify the downlink bridge, enter **bridge show**.

```
zSH> bridge show
```

Type	Orig		Physical	Bridge	St	Table Data
	VLAN/SLAN	VLAN/SLAN				
dwn-dat		300	1/6/1/0/eth	1-6-1-0-eth/bridge	UP	

1 Bridge Interfaces displayed

The **vlanId** parameter is set to 555 and will be stripped on the downstream and inserted on the upstream.

Tagged downlink bridges on Active Ethernet

You configure a downlink bridge as *tagged* when a VLAN ID is expected or needed in the downstream configuration.

Designating a downlink bridge as *tagged* means that the VLAN ID is not stripped out of the Ethernet packet, and is delivered intact to a device expecting traffic with the designated VLAN ID. The VLAN ID remains unchanged when traveling in the upstream direction.

Configuring an Active Ethernet tagged downlink VLAN bridge

- 1 Create a *tagged* downlink bridge with a VLAN ID.

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 300 tagged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-300/bridge
```

- 2 To display the tagged downlink bridge, enter **bridge show**.

```
zSH> bridge show
```

Type	Orig		Physical	Bridge	St	Table Data
	VLAN/SLAN	VLAN/SLAN				
dwn-dat	Tagged	300	1/6/1/0/eth	1-6-1-0-eth-300/bridge	UP	

1 Bridge Interfaces displayed

The VLAN ID parameter is set to 300. Since the downlink bridge is *tagged* the VLAN ID remains in the Ethernet packet and stays intact in both directions.

Deleting a downlink bridge

Delete a downlink bridge when necessary.

```
zSH> bridge delete 1-6-1-0-eth/bridge vlan 300
1-6-1-0-eth/bridge delete complete
```

TLS bridges with VLAN ID

This section describes TLS bridge configurations including:

- [TLS bridges, page 246](#)
- [TLS bridge parameters floodUnknown and floodMulticast, page 247](#)

TLS bridges learn MAC addresses and forward packets to learned destinations. Broadcasts and unknown unicasts are flooded out all interfaces except the ingress interface. Packets entering the system on a TLS interface have their source MAC addresses learned and associated with the interface so that frames from the network that come in on other TLS bridges in the VLAN can be sent to the correct interface.

TLS bridges

TLS is a symmetrical bridge and can only be used with other TLS bridges.

Creating a TLS bridge configuration

- 1 Create a TLS bridge on the MXK Active Ethernet card.

```
zSH> bridge add 1-13-6-0/eth tls vlan 900
Adding bridge on 1-13-6-0/eth
Created bridge-interface-record 1-13-6-0-eth/bridge
Bridge-path added successfully
```

TLS bridges automatically create a bridge path on the first instance of the VLAN ID.

- 2 Create a TLS bridge on the uplink card.

```
zSH> bridge add 1-a-2-0/eth tls vlan 900
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2/bridge
```

- 3 View the TLS bridges.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
tls            900        1/13/6/0/eth      1-13-6-0-eth/bridge  DWN
tls            900        1/a/2/0/eth       ethernet2/bridge     DWN
2 Bridge Interfaces displayed
```

Deleting the TLS bridge configuration

- 1 Delete the bridge on the uplink card.

```
zSH> bridge delete ethernet2/bridge vlan 900
ethernet2/bridge delete complete
```

- 2 Delete the bridge on the Active Ethernet card.

```
zSH> bridge delete 1-13-6-0-eth/bridge vlan 900
Bridge-path deleted successfully
1-13-6-0-eth/bridge delete complete
```

TLS bridge parameters **floodUnknown** and **floodMulticast**

TLS bridges can provide VPN-like services with the **floodUnknown** and **floodMulticast** parameters that allow the MXK to forward unknown traffic to all bridge interfaces within the VLAN.

***floodUnknown** parameter*

The **floodUnknown** parameter provides the ability to flood unknown unicast destination frames with unknown unicast MAC addresses to all interfaces on the VLAN. One case where this may need to be done is when voice packets are flooded out on the VLAN to see if there is an interface that will respond.

When the **floodUnknown** parameter is set to *true*, the MXK forwards (floods) frames with unknown unicast MAC addresses to all interfaces on the VLAN. The **learnUnicast** parameter is set to *true*. If a interface responds to a flooded packet, the address is learned, and that packet does not need to be flooded again.

When this parameter is set to *false*, the MXK discards frames with an unknown unicast MAC addresses. Frames that do not find a match in the forwarding table are discarded.

For TLS bridges, the default setting for these parameters is *true*. For uplink downlink, and intralink bridges, the default setting for these parameters is *false*.

This example shows that the **floodUnknown** and **learnUnicast** default settings are set to true on a TLS bridge.

```
zSH> bridge add 1-13-1-0/eth tls vlan 500
Adding bridge on 1-13-1-0/eth
Created bridge-interface-record 1-13-1-0-eth/bridge
Bridge-path added successfully

zSH> get bridge-interface-record 1-13-1-0-eth/bridge
bridge-interface-record 1-13-1-0-eth/bridge
vpi: -----> {0}
vci: -----> {0}
vlanId: -----> {500}
stripAndInsert: -----> {true}
customARP: -----> {false}
filterBroadcast: -----> {false}
learnIp: -----> {false}
learnUnicast: -----> {true}
maxUnicast: -----> {100}
learnMulticast: -----> {false}
forwardToUnicast: -----> {true}
forwardToMulticast: -----> {false}
```

```

forwardToDefault: -----> {false}
bridgeIfCustomDHCP: -----> {false}
bridgeIfIngressPacketRuleGroupIndex: -> {0}
vlanIdCOS: -----> {0}
outgoingCOSOption: -----> {disable}
outgoingCOSValue: -----> {0}
s-tagTPID: -----> {0x8100}
s-tagId: -----> {0}
s-tagStripAndInsert: -----> {true}
s-tagOutgoingCOSOption: -----> {s-tagdisable}
s-tagIdCOS: -----> {0}
s-tagOutgoingCOSValue: -----> {0}
mcastControlList: -----> {}
maxVideoStreams: -----> {0}
isPPPoA: -----> {false}
floodUnknown: -----> {true}
floodMulticast: -----> {true}
bridgeIfEgressPacketRuleGroupIndex: --> {0}
bridgeIfTableBasedFilter: -----> {NONE(0)}
bridgeIfDhcpLearn: -----> {NONE(0)}
mvrVlan: -----> {0}
vlan-xlate-from: -----> {0}
slan-xlate-from: -----> {0}

```

floodMulticast parameter

The **floodMulticast** parameter allows the MXK to flood all multicast traffic received on a bridge out to all other ports in the VLAN. Multicast traffic in this case usually means multicast traffic that is not for video. For example, many routing protocols are found in multicast packets. This is useful for architectures where the MXK is acting as an aggregation point with no user interfaces. By default, this parameter is set to *true* on TLS bridges and *false* on uplink and downlink bridges.

When set to *true*, this parameter causes all multicast frames to be forwarded out all of the bridge interfaces within the VLAN, except the interface where the multicast was received.

To view the setting for this parameter, enter get **bridge-interface-record**:

```

zSH> bridge add 1-13-1-0/eth tls vlan 500
Adding bridge on 1-13-1-0/eth
Created bridge-interface-record 1-13-1-0-eth/bridge
Bridge-path added successfully

zSH> get bridge-interface-record 1-13-1-0-eth/bridge
bridge-interface-record 1-13-1-0-eth/bridge
vpi: -----> {0}
vci: -----> {0}
vlanId: -----> {500}
stripAndInsert: -----> {true}
customARP: -----> {false}
filterBroadcast: -----> {false}
learnIp: -----> {false}

```

```

learnUnicast: -----> {true}
maxUnicast: -----> {100}
learnMulticast: -----> {false}
forwardToUnicast: -----> {true}
forwardToMulticast: -----> {false}
forwardToDefault: -----> {false}
bridgeIfCustomDHCP: -----> {false}
bridgeIfIngressPacketRuleGroupIndex: -> {0}
vlanIdCOS: -----> {0}
outgoingCOSOption: -----> {disable}
outgoingCOSValue: -----> {0}
s-tagTPID: -----> {0x8100}
s-tagId: -----> {0}
s-tagStripAndInsert: -----> {true}
s-tagOutgoingCOSOption: -----> {s-tagdisable}
s-tagIdCOS: -----> {0}
s-tagOutgoingCOSValue: -----> {0}
mcastControlList: -----> {}
maxVideoStreams: -----> {0}
isPPPoA: -----> {false}
floodUnknown: -----> {true}
floodMulticast: -----> {true}
bridgeIfEgressPacketRuleGroupIndex: --> {0}
bridgeIfTableBasedFilter: -----> {NONE(0)}
bridgeIfDhcpLearn: -----> {NONE(0)}
mvrVlan: -----> {0}
vlan-xlate-from: -----> {0}
slan-xlate-from: -----> {0}

```

Wire bridge configuration

This section describes:

- [Nonlearning and learning wire bridges, page 250](#)
- [GPON wire bridge Q-in-Q-in-Q encapsulation, page 253](#)

Nonlearning and learning wire bridges

When configuring wire bridges, the VLAN ID used on the two wire bridge interfaces is reserved for the entire device and cannot be used again. Wire bridges are confined to two bridge interfaces on a VLAN ID. Additional bridge interfaces on the VLAN ID cannot be added.

Default wire bridge behavior is nonlearning with broadcasts and unicasts forwarded to the second wire bridge.



Note: Zhone recommends using non-learning wire bridges as they do not require MAC table forwarding space. Packets are forwarded between the ingress and egress ports based purely on VLAN membership

There is *one* exception to the TWO end points rule. Wire bridges can be configured between a line card and two Ethernet ports on an EAPS transit node.

For example:

```
zSH> bridge add 1-a-2-0/eth wire vlan 100 tagged
```

```
zSH> bridge add 1-b-2-0/eth wire vlan 100 tagged
```

```
zSH> bridge add 1-7-3-509/gponport wire vlan 100 tagged
```



Note: This wire bridge configuration is only valid on EAPS transit nodes.

If learning behavior is required on the wire bridge, the wire bridge can be configured with the enable learn unicast feature by entering the keyword *learning*. The learn unicast feature can then be disabled by entering the keyword *nolearning* with the **bridge modify** command.

Configuring a default wire bridge

- 1 Create the first wire bridge interface with VLAN ID.

```
zSH> bridge add 1-a-9-0/eth wire vlan 999
Adding bridge on 1-a-9-0/eth
Created bridge-interface-record ethernet9/bridge
Bridge-path added successfully
```

- 2 Create the second wire bridge interface with the same VLAN ID.

```
zSH> bridge add 1-6-1-0/eth wire vlan 999
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth/bridge
```

3 View the wire bridges.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
wre           999      1/6/1/0/eth  1-6-1-0-eth/bridge  UP  No Learning
wre           999      1/a/9/0/eth  ethernet9/bridge    DWN No Learning
2 Bridge Interfaces displayed
```

If a VLAN ID is used for two wire bridges, the system prevents that VLAN ID from being used again.

```
zSH> bridge add 1-6-2-0/eth wire vlan 999
Error: Wire bridge on a given s/vlan exceeds the limit on physical
Unable to create bridge-interface-record 1-6-2-0-eth/bridge
```

Configuring learning wire bridges

When needed, wire bridges can be configured as learning wire bridges.



Note: Wire bridges with learning are valid only on GPON, Active Ethernet, and EFM SHDSL in the upstream and downstream direction.

1 Create the first wire bridge with VLAN ID and the keyword *learning*.

```
zSH> bridge add 1-a-2-0/eth wire vlan 400 learning
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2/bridge
Bridge-path added successfully
```

2 Create the second wire bridge interface with the same VLAN ID.

```
zSH> bridge add 1-6-2-0/eth wire vlan 400 learning
Adding bridge on 1-6-2-0/eth
Created bridge-interface-record 1-6-2-0-eth/bridge
```

3 Verify the bridges.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
wre           400      1/6/2/0/eth  1-6-2-0-eth/bridge  DWN
wre           400      1/a/2/0/eth  ethernet2/bridge    DWN
2 Bridge Interfaces displayed
```

Changing learning wire bridges to nolearning wire bridges

When needed, change the learning wire bridges to nolearning wire bridges.

1 Modify the first wire bridge using the *nolearning* keyword.

```
zSH> bridge modify ethernet2/bridge nolearning
ethernet2/bridge has been modified
```

2 Modify the second wire bridge using the *nolearning* keyword.

```
zSH> bridge modify 1-6-2-0-eth/bridge nolearning
1-6-2-0-eth/bridge has been modified
```

3 Verify the changes to the bridge interfaces.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
wre           400    1/6/2/0/eth  1-6-2-0-eth/bridge  DWN No Learning
wre           400    1/a/2/0/eth  ethernet2/bridge    DWN No Learning
2 Bridge Interfaces displayed
```


GPON wire bridge Q-in-Q-in-Q encapsulation

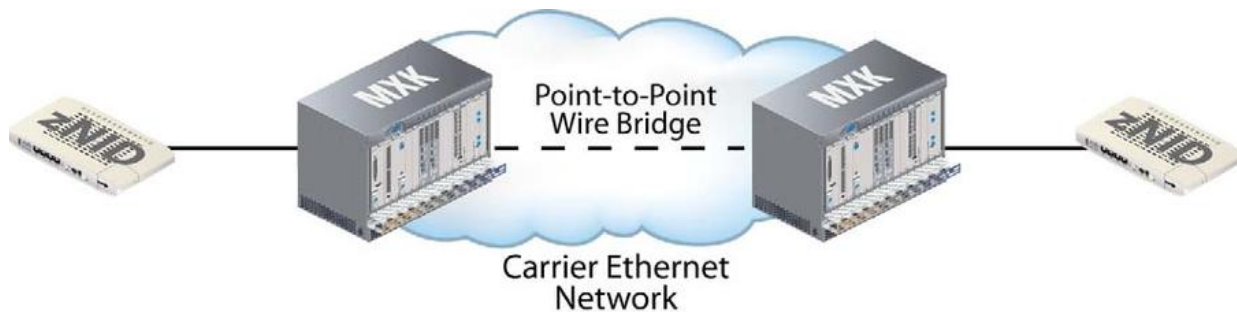
The MXK supports Q-in-Q-in-Q in an *staged* configuration on GPON wire bridges.

The rules for Q-in-Q-in-Q on GPON are:

- Wire bridges are now independent of the TLS bridge type.
- Q-in-Q-in-Q is *only* for GPON wire bridge configurations.
- The network facing and subscriber facing wire bridges must be s-tagged using VLAN 0 and SLAN ID with the **bridge add** command.
- The subscriber zNID must be configured in s-tagged VLAN mode with the desired TPID.

Q-in-Q-in-Q on GPON wire bridges provides the mechanism for VLAN encapsulation of subscriber traffic by adding additional tags to Ethernet frames as shown in [Figure 31](#).

Figure 31: Wire bridge configuration for Q-in-Q-in-Q



Configuring a GPON wire bridge for Q-in-Q-in-Q

- 1 Configure the subscriber facing GPON wire bridge with VLAN 0, SLAN ID, and keyword *staged*.

```
zSH> bridge add 1-5-1-6/gpononu gem 506 gtp 1 wire vlan 0 slan 600 staged
Adding bridge on 1-5-1-6/gpononu
Created bridge-interface-record 1-5-1-506-gponport-0-600/bridge
Bridge-path added successfully
```

- 2 Configure the network facing GPON wire bridge with VLAN 0, SLAN ID, and keyword *staged*.

```
zSH> bridge add 1-a-5-0/eth wire vlan 0 slan 600 staged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-0-600/bridge
```

Verify the bridges.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
wre           ST    0/600    1/5/1/6/gpononu    1-5-1-506-gponport-0-600/bridge  DWN No Learning
wre           ST    0/600    1/a/5/0/eth        ethernet5-0-600/bridge  UP  No Learning
2 Bridge Interfaces displayed
```

Q-in-Q on bridges (VLAN IDs and SLAN IDs)

The MXK supports two ways of configuring Q-in-Q in bridging. The first way uses the *tagged* variable and the second way uses the *staged* variable. Some MXK bridging configurations are from an *staged* bridge to a *tagged* bridge (see [Tagged downlink bridge to staged uplink bridge \(SLAN promotion\) on page 255](#)), or from a *staged* bridge to a *staged* bridge (see [Uplink staged bridge to downlink staged bridge on page 254](#)).

Overview of Q-in-Q (VLAN/SLAN)

The IEEE 802.1Q-in-Q VLAN tagging expands the VLAN space in the Ethernet frame to support the tagging of previously tagged packets. This second tag (SLAN) creates a "double-tagged" Ethernet frame.

In double-tagged or *staged* configurations, there is a VLAN ID and an SLAN ID. When the bridge interface with both a VLAN ID and an SLAN ID is configured to *tagged* the VLAN ID *is not* stripped and inserted and the SLAN ID *is* stripped and inserted. On the downlink this means that the VLAN ID is passed down, but the SLAN ID is not. The SLAN ID is stripped out for the egress traffic, and inserted back for the ingress traffic.

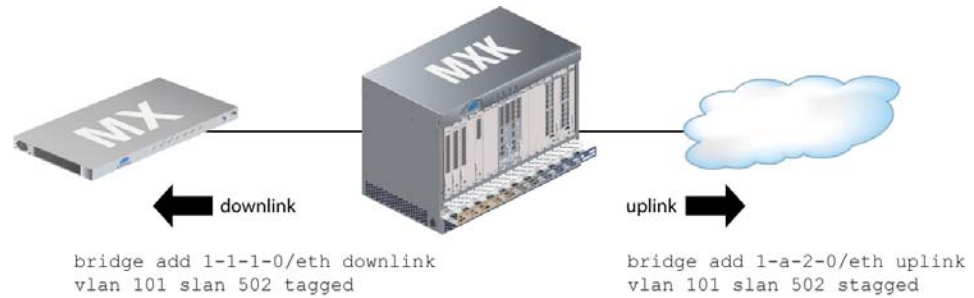
When the bridge interface with both a VLAN ID and an SLAN ID is configured to *staged*, neither the VLAN ID nor the SLAN ID are stripped and inserted. Both the VLAN ID and the SLAN ID are passed to the downstream device.

The MXK also supports setting CoS values in the Ethernet SLAN headers for bridged packets. This service enables the user to assign a service level or class of service (CoS) to an Ethernet SLAN that is transported across a uplink, intralink, or downlinked s-tagged bridge. The configured CoS level specifies the packet priority and queueing methods used to transport the packet through the Ethernet network. The MXK sets and preserves the CoS settings to ensure these settings are passed to other Ethernet devices in the network for QoS processing. See [Shaping Traffic: Class of Service Queuing on page 406](#).

Uplink staged bridge to downlink staged bridge

[Figure 32](#) describes an *staged* downlink to *staged* uplink bridging configuration.

Figure 32: tagged to tagged uplink downlink configuration



Configuring an tagged bridge on the downlink and an tagged bridge on the uplink

- 1 Create an *tagged* uplink bridge with VLAN ID and SLAN ID.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 102 slan 502 tagged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-102-502/bridge
Bridge-path added successfully
```

Designating *tagged* passes both the VLAN ID and SLAN ID to the network.

- 2 Create an *tagged* downlink bridge with the same SLAN ID and a VLAN ID and the uplink bridge.

```
zSH> bridge add 1-1-6-0/eth downlink-data vlan 102 slan 502 tagged
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth-102-502/bridge
```

Designating the downlink bridge as *tagged* passes both the VLAN ID and the SLAN ID to the downstream device.

- 3 Verify the bridge:

```
zSH> bridge show
```

Type	Orig	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
dwn-dat	ST	102/502	1/1/6/0/eth	1-1-6-0-eth-102-502/bridge	DWN			
upl	ST	102/502	1/a/5/0/eth	ethernet5-102-502/bridge	DWN	S	SLAN 502	VLAN 102 default

2 Bridge Interfaces displayed

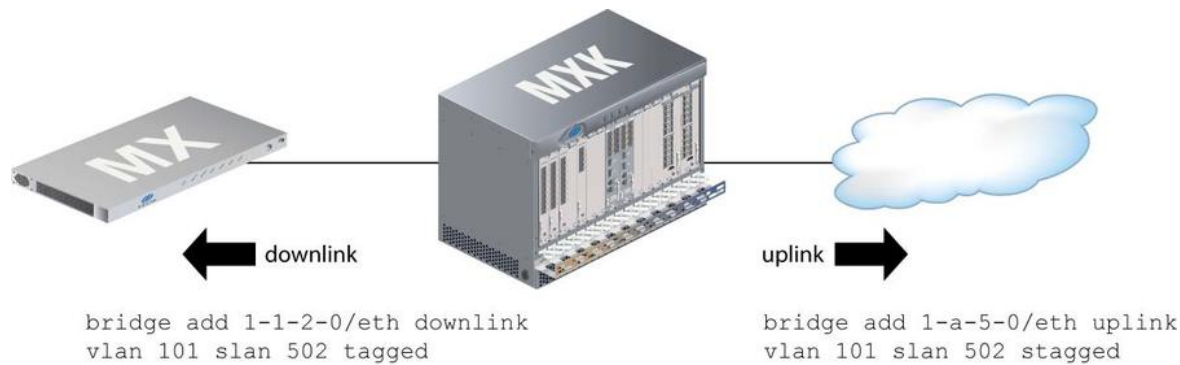
Tagged downlink bridge to tagged uplink bridge (SLAN promotion)

Figure 33 shows an example of using Q-in-Q (SLAN IDs) on both the uplink and the downlink bridge, but designating *tagged* on the downlink bridge and *tagged* on the uplink bridge.

In this case, designating the downlink bridge as *tagged* causes the SLAN ID to be stripped as it passes to the downstream device, and re-inserted when traveling in the upstream direction. The VLAN ID remains in both directions.

This type of configuration allows a downstream device such as a MALC to receive the VLAN ID and not the SLAN ID. [Figure 33](#) shows a tagged downlink and staged uplink bridging configuration.

Figure 33: Tagged downlink and staged uplink example



Configuring a staged uplink and tagged downlink bridge

This configuration will create a downlink bridge that strips out the SLAN ID on the downlink and re-inserts the SLAN ID when traveling to the uplink and an uplink that sends both the VLAN ID and the SLAN ID to the network.

- 1 Create an *staged* uplink bridge with a VLAN ID and a SLAN ID.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 101 slan 501 staged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-101-501/bridge
Bridge-path added successfully
```

Designating the uplink bridge as *staged* does not strip or insert the either the VLAN ID or the SLAN ID.

- 2 Create a *tagged* downlink bridge with an SLAN ID 501 and a VLAN ID 101 to match the uplink bridge.

```
zSH> bridge add 1-1-6-0/eth downlink-data vlan 101 slan 501 tagged
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth-101/bridge
```

Designating the downlink bridge as *tagged* strips the SLAN ID on the way to the downstream device and re-inserts the SLAN ID on the way to the uplink. The VLAN ID remains in both directions.

- 3 To verify the bridges enter:

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
dwn-dat      Tg  101/501  1/1/6/0/eth  1-1-6-0-eth-101/bridge  DWN
upl          ST  102/502  1/a/5/0/eth  ethernet5-102-502/bridge  DWN S SLAN 502 VLAN 102 default
2 Bridge Interfaces displayed
```

untagged downlink bridge to staged uplink bridge (double-promotion)



Note: Only EFM SHDSL and ADSL cards support double-promotion.

In certain cases it may be useful to create an *staged* uplink bridge with *untagged* downlink bridges. For example, when the downlink bridges are connected to DSL modems that do not recognize VLAN or SLAN IDs and the when the network device is expecting both a VLAN ID and an SLAN ID.

Creating an untagged to staged bridge configuration (double-promotion) on EFM SHDSL

- 1 Verify the EFM SHDSL bond groups.

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
12	30	efmbond	OOS	bond-0030	-

- 2 Create the *untagged* downlink bridge on the bond group.

```
zSH> bridge add bond-0030/efmbond downlink-data vlan 101 slan 501 untagged
```

Adding bridge on bond-0030/efmbond

Created bridge-interface-record bond-0030-efmbond/bridge



Note: The *downlink* variable *must* be designated for a uplink/downlink bridging configuration. Otherwise the subscriber facing bridge defaults to the type TLS.

- 3 Create the *staged* uplink bridge.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 101 slan 501 staged
```

Adding bridge on 1-a-5-0/eth

Created bridge-interface-record ethernet5-101-501/bridge

Bridge-path added successfully

- 4 Verify the bridges.

```
zSH> bridge show
```

Type	Orig	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-dat		101/501	1/12/30/0/efmbond	bond-0030-efmbond/bridge	DWN	
upl	Tagged 8		1/a/2/0/eth	ethernet2-8/bridge	DWN S	VLAN 8 default

2 Bridge Interfaces displayed

Delete the uplink and downlink bridges

Deleting the uplink and downlink bridge

1 Delete the uplink bridge.

```
zSH> bridge delete ethernet5-101-501/bridge vlan 101 slan 501
Bridge-path deleted successfully
ethernet5-101-501/bridge delete complete
```

2 Delete the downlink bridge.

```
zSH> bridge delete bond-0030-efmbond/bridge
bond-0030-efmbond/bridge delete complete
```

Turn off Q-in-Q for the entire MXK system

Setting the **options** parameter in the **system 0** profile to *cvlanonly* turns off the ability to configure bridges with SLAN IDs.

```
zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {Zhone Global Services and Support 7195 Oakport
Street Oakland Ca. (877) Zhone20 (946-6320) Fax (510) 777-7113
support@zhone.com}:
sysname: -----> {Zhone MxK}:
syslocation: -----> {Oakland}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {false}:
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: ---> {false}:
configsyncpriority: ---> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: ---> {}:
configsyncstatus: -----> {syncinitializing}:
configsyncuser: -----> {}:
configsyncpasswd: -----> {** private **}: ** read-only **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {0.0.0.0}:
alternateipaddress: ---> {0.0.0.0}:
countryregion: -----> {us}:
primaryclocksource: ---> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
```

```

secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {NONE(0)}: cvlanonly
reservedVlanIdStart: --> {0}:
reservedVlanIdCount: --> {0}:
snmpVersion: -----> {snmpv2}:
persistentLogging: ----> {disabled}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Please reboot the system for cvlanonly change to take effect.
Record updated.
zSH> systemreboot
Do you want to reboot the system? (yes or no) [no] yes
Do you want to exit from this request? (yes or no) [yes] no
Are you sure? (yes or no) [no] yes

```

If you now attempt to create a bridge with an SLAN ID, you will get the following error message:

```

zSH> bridge add 1-13-6-0/eth downlink vlan 777 slan 20
Adding bridge on 1-13-6-0/eth
Error: slan must be 0 for untag interface.

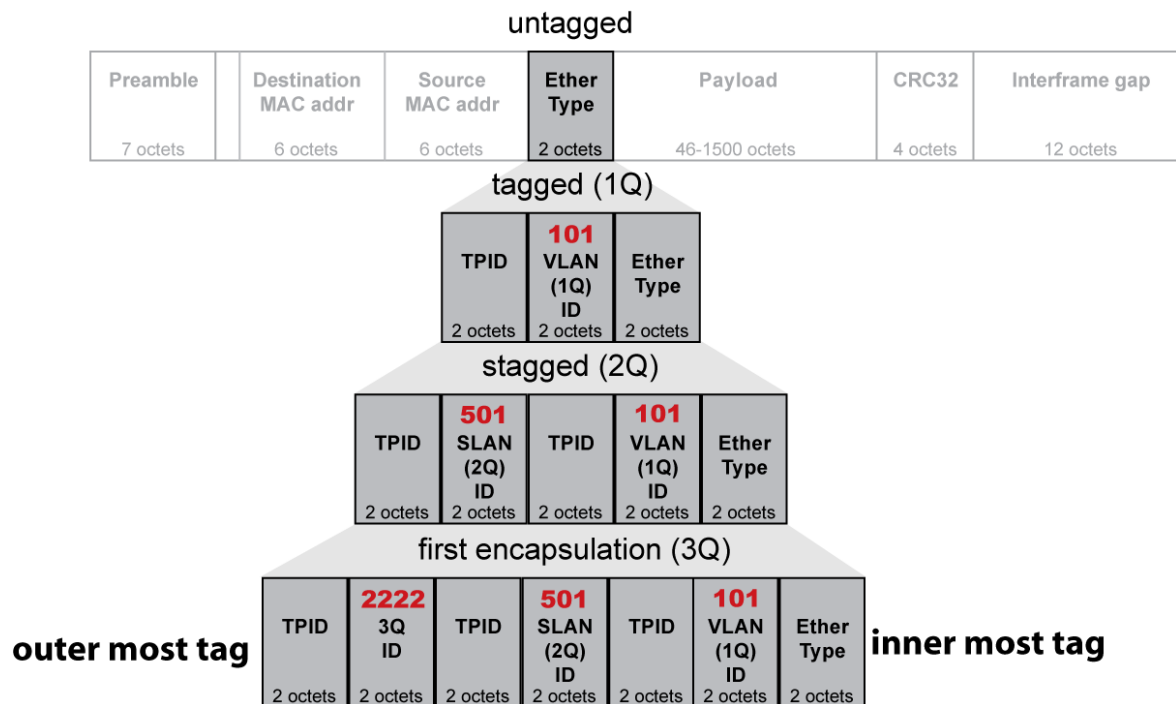
```

Q-in-Q-in-Q (VLAN IDs, SLAN IDs and packet rules) on bridges

Q-in-Q-in-Q overview

The MXK implements Q-in-Q-in-Q with packet rules on *staged* TLS bridges. The packet rule promotes the third tag by inserting the tag to the network and stripping the tag to the access. See [Filters for MXK bridges \(packet-rule-record\)](#), page 313 for more information on creating packet rules.

Figure 34: IP header changes for Q-in-Q-in-Q



Rules for applying a **packet-rule record** for Q-in-Q-in-Q:

- The line cards for access on the MXK that support Q-in-Q-in-Q are:
 - MXK-AEX20-FE/GE (single-slot)
 - MXK-AEX20-FE/GE-CSFP
 - The uplink cards on the MXK that support Q-in-Q-in-Q are:
 - MXK MXK-UPLINK-2X10G-8X1GE
 - MXK MXK-UPLINK-8X1GE
 - MXK-UPLINK-4X1GE
 - TLS is the only bridge type that supports the Q-in-Q-in-Q packet rules.
 - Both the access facing and the network facing TLS bridges must be *staged* with matching VLAN and SLAN IDs.
 - Valid VLAN and SLAN IDs are between 1-4090. Wildcard VLAN ID 0 is supported. Wildcards are not supported on the SLAN ID.
 - The packet rules *promotefirstencapsulationvlan* and *filterfirstencapsulationvlan* cannot exist in the same **packet-rule-record** group.
- See [Filters for MXK bridges \(packet-rule-record\)](#), page 313 for information on creating packet rules.
- The packet rules for Q-in-Q-in-Q can only be assigned on the ingress of the bridge interface.

- *promotefirstencapsulationvlan* can only be used on an access port.
- *filterfirstencapsulationvlan* can only be used on a uplink port.

Configure an access TLS bridge for Q-in-Q-in-Q

For this Q-in-Q-in-Q configuration, the outer tag will be stripped going to the access TLS bridge and inserted (promoted) to the network TLS bridge.

Configuring subscriber facing TLS bridges for Q-in-Q-in-Q

- 1 Create the *promotefirstencapsulationvlan* **packet-rule-record** to define the outer VLAN ID (third tag) for the access facing TLS bridge that will be promoted to the network.

Enter the VLAN ID, the TPID, and CoS for the packet rule from the CLI with the **rule add** command.

```
zSH> rule add promotefirstencapsulationvlan 1/1 vlanid 2222 tpid 0x8100 cos 7
Created packet-rule-record 1/1 (promotefirstencapsulationvlan)
```

- 2 Verify the packet rule.

```
zSH> get packet-rule-record 1/1
packet-rule-record 1/1
packetRuleType: ---> {promotefirstencapsulationvlan}
packetRuleValue: --> {2222}
packetRuleValue2: -> {0x8100}
packetRuleValue3: -> {7}
packetRuleValue4: -> {}
packetRuleValue5: -> {}
packetRuleValue6: -> {}
packetRuleValue7: -> {}
```

```
zSH> rule show
```

Group/Member	Type	Value(s)
Default BSD dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30
		auto-enable-interval (def) 300 600 1200
Default BSD tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30
		auto-enable-interval (def) 300 600 1200
1/1 promotefirstencapsulationvlan		vlanid 2222 tpid 0x8100 cos 7

3 record(s) found

- 3 Create the access facing *staged* TLS bridge with VLAN ID and SLAN ID, and apply packet rule 1 for Q-in-Q-in-Q.

```
zSH> bridge add 1-6-1-0/eth tls vlan 101 slan 501 staged ipktrule 1
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-101-501/bridge
Bridge-path added successfully
```

- 4 Verify the bridge.

```
zSH> bridge show
```

Type	Orig		Physical	Bridge	St	Table	Data
	VLAN/SLAN	VLAN/SLAN					
tls	ST	101/501	1/6/1/0/eth	1-6-1-0-eth-101-501/bridge	UP	Promote	encVln 2222
1 Bridge Interfaces displayed							

5 Locate interfaces configured with the packet rule if necessary.

```
zSH> rule showuser
Group/Member                                Type  IfIndex  IfAddr
1/1      promotefirstencapsulationvlan      1036  1-6-1-0-eth-101-501/bridge (ingress)
1 record(s) found
```

Configure a network facing TLS bridge for Q-in-Q-in-Q

For this Q-in-Q-in-Q configuration, the outer tag will be stripped going to the access TLS bridge and inserted (promoted) to the network TLS bridge.

Configuring network facing TLS bridges for Q-in-Q-in-Q

1 Create the *filterfirstencapsulationvlan* packet-rule-record for the network facing TLS bridge.

The VLAN ID for the outer tag must match the VLAN ID of the *promotefirstencapsulationvlan* packet-rule-record.

```
zSH> rule add filterfirstencapsulationvlan 2/1 vlanid 2222 tpid 0x8100
Created packet-rule-record 2/1 (filterfirstencapsulationvlan)
```

2 Verify the packet rule.

```
zSH> get packet-rule-record 2/1
packet-rule-record 2/1
packetRuleType: ---> {filterfirstencapsulationvlan}
packetRuleValue: --> {2222}
packetRuleValue2: -> {0x8100}
packetRuleValue3: -> {}
packetRuleValue4: -> {}
packetRuleValue5: -> {}
packetRuleValue6: -> {}
packetRuleValue7: -> {}
```

```
zSH> rule show
Group/Member                                Type  Value(s)
-----
Default BSD dwn (0/1)                      bridgestormdetect discard+alarm+block pps 30 cs 30
                                           auto-enable-interval (def) 300 600 1200
Default BSD tls/wire (0/2)                  bridgestormdetect discard+alarm+block pps 100 cs 30
                                           auto-enable-interval (def) 300 600 1200
1/1      promotefirstencapsulationvlan      vlanid 2222 tpid 0x8100 cos 7
2/1      filterfirstencapsulationvlan      vlanid 2222 tpid 0x8100
4 record(s) found
```

- 3 Create the network facing *staged* TLS bridge with VLAN ID and SLAN ID that match the subscriber facing bridge, and apply packet rule 2 for Q-in-Q-in-Q.

```
zSH> bridge add 1-a-2-0/eth tls vlan 101 slan 501 staged ipktrule 2
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-101-501/bridge
```

- 4 Verify the bridge.

```
zSH> bridge show
```

Type	Orig	VLAN/SLAN	Physical	Bridge	St	Table	Data
tls	ST	101/501	1/6/1/0/eth	1-6-1-0-eth-101-501/bridge	UP	Promote	encVln 2222
tls	ST	101/501	1/a/2/0/eth	ethernet2-101-501/bridge	DWN	Filter	encVlan 2222

6 Bridge Interfaces displayed

- 5 Locate interfaces configured with the packet rule if necessary.

```
zSH> rule showuser
```

Group/Member	Type	IfIndex	IfAddr
1/1	promotefirstencapsulationvlan	1036	1-6-1-0-eth-101-501/bridge (ingress)
2/1	filterfirstencapsulationvlan	1037	ethernet2-101-501/bridge (ingress)

2 record(s) found

Bridges using VLAN 0

On the MXK, VLAN 0 functions as a wildcard that will recognize all VLAN IDs but can only be used in conjunction with an SLAN ID. You can designate VLAN 0 on uplink, downlink, TLS, and intralink bridges. Any bridge configuration using VLAN 0 can be designated either *tagged* or *staged* depending on the bridging behavior desired on the subscriber facing side. For SHDSL EFM and ADSL cards, *untagged* VLAN ID/SLAN ID is supported with promotion towards the network.

Possible bridging configuration behaviors for VLAN 0

Each of the following bridging configuration examples all assume an uplink configuration of VLAN 0 SLAN x *staged*:

- The network facing bridge is *staged* and the subscriber facing bridge has VLAN x and SLAN x *staged*.
- The network facing bridge is *staged* and the subscriber facing bridge has VLAN x and SLAN x *tagged*.
- The network facing bridge is *staged* and the subscriber facing bridge has VLAN 0 SLAN x *staged*.
- The network facing bridge is *staged* and the subscriber facing bridge has VLAN x and SLAN x *untagged*. (Promotion is supported only on EFM SHDSL and ADSL cards.)

Uplink bridges with VLAN 0 SLAN ID staged configuration cases

Creating an staged uplink bridge with VLAN 0 and SLAN ID and downlink bridge with VLAN ID and SLAN ID staged

- 1 Create the *staged* uplink with VLAN 0 and SLAN ID.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 0 slan 501 staged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-0-501/bridge
Bridge-path added successfully
```

- 2 Create the *staged* downlink bridge with a designated VLAN ID and SLAN ID.

```
zSH> bridge add 1-1-1-0/eth downlink-data vlan 100 slan 101 staged
Adding bridge on 1-1-1-0/eth
Created bridge-interface-record 1-1-1-0-eth-100-101/bridge
```

Verify the bridges.

```
zSH> bridge show
```

Type	Orig	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-dat		ST	100/101	1/1/1-0/eth	1-1-1-0-eth-100-101/bridge	DWN	
upl		ST	0/501	1/a/5/0/eth	ethernet5-0-501/bridge	DWN S	SLAN 501 VLAN 0 default

2 Bridge Interfaces displayed

Creating an staged uplink bridge with VLAN 0 and SLAN ID and downlink bridge with VLAN ID and SLAN ID tagged

- 1 Create the *staged* uplink with VLAN 0 and SLAN ID.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 0 slan 501 staged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-0-501/bridge
Bridge-path added successfully
```

- 2 Create the *tagged* downlink bridge with a designated VLAN ID and SLAN ID.

```
zSH> bridge add 1-1-1-0/eth downlink-data vlan 200 slan 501 tagged
Adding bridge on 1-1-1-0/eth
Created bridge-interface-record 1-1-1-0-eth-200/bridge
```

Verify the bridges.

```
zSH> bridge show
```

Type	Orig	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-dat		Tg	200/501	1/1/1-0/eth	1-1-1-0-eth-200/bridge	DWN	
upl		ST	0/501	1/a/5/0/eth	ethernet5-0-501/bridge	DWN S	SLAN 501 VLAN 0 default

2 Bridge Interfaces displayed

Creating an staged uplink bridge with VLAN 0 and SLAN ID and downlink bridge with VLAN 0 and SLAN ID staged

In situations where a business subscriber uses many internal VLAN IDs that the network service provider does not care about, you can configure the downlink bridge with VLAN ID 0 and an SLAN ID. The SLAN ID will be recognized going out to the network and all VLAN IDs will be passed down to the business subscriber.

- 1 Create the *staged* uplink with VLAN 0 and SLAN ID.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 0 slan 501 staged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-0-501/bridge
Bridge-path added successfully
```

All VLAN IDs will be passed to the network on SLAN 501.

- 2 Create the *staged* downlink bridge with VLAN 0 and specify the SLAN ID.

```
zSH> bridge add 1-1-1-0/eth downlink-data vlan 0 slan 501 staged
Adding bridge on 1-1-1-0/eth
Created bridge-interface-record 1-1-1-0-eth-0-501/bridge
```

All VLAN IDs will be passed downstream on SLAN 501.

Verify the bridges.

```
zSH> bridge show
```

Type	Orig	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-dat	ST	0/501	1/1/1-0/eth	1-1-1-0-eth-0-501/bridge	DWN	
upl	ST	0/501	1/a/5-0/eth	ethernet5-0-501/bridge	DWN S	SLAN 501 VLAN 0 default

2 Bridge Interfaces displayed

Creating an staged uplink bridge with VLAN 0 and SLAN ID and downlink bridge with VLAN ID and SLAN ID untagged

Note: This configuration can only be performed on EFM SHDSL or ADSL cards.

- 1 Create the *staged* uplink with VLAN 0 and SLAN ID.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 0 slan 501 staged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-0-501/bridge
Bridge-path added successfully
```

All VLAN IDs will be passed to the network on SLAN 501.

2 Create the *untagged* downlink with VLAN ID and SLAN ID.

```
zSH> bridge add 1-10-1-0/adsl vc 0/35 td 1 downlink-data vlan 300 slan 501 untagged
Adding bridge on 1-10-1-0/adsl
Created bridge-interface-record 1-10-1-0-adsl-0-35/bridge
```

View the bridges.

```
zSH> bridge show
```

Type	Orig	VLAN/SLAN	Physical	Bridge	St	Table	Data
dwn-dat		300/501	1/10/1/0/adsl	1-10-1-0-adsl-0-35/bridge	DWN		
upl	ST	0/501	1/a/5/0/eth	ethernet5-0-501/bridge	UP	S	SLAN 501 VLAN 0 default

2 Bridge Interfaces displayed

Deleting the uplink and downlink bridges with VLAN 0

If necessary, delete the uplink and downlink bridges.

1 Delete the uplink bridge.

```
zSH> bridge delete ethernet5-0-501/bridge vlan 0 slan 501
Bridge-path deleted successfully
ethernet5-0-501/bridge delete complete
```

2 Delete the downlink bridge.

```
zSH> bridge delete 1-10-1-0-adsl-0-35/bridge
1-10-1-0-adsl-0-35/bridge delete complete
```

MXK bridging configuration with VLAN 0 on TLS bridges for multi-point connections

In bridging configurations where multi-point connections are needed, you can configure TLS bridges with VLAN 0 and the same SLAN ID. A multi-point connection is two or more connections for the same SLAN ID facing the subscriber. The TLS bridge facing the subscriber is *tagged*. This means the SLAN ID is stripped out to the subscriber and inserted to the network. The TLS bridge to the network is *staged*, keeping both the VLANs and the SLAN ID. The network device will recognize the SLAN ID, i.e. the outer tag.

Creating TLS bridges for a multi-point connection

First create the TLS bridge with VLAN 0 and the SLAN ID on the network facing Ethernet port, then create the TLS bridges on the subscriber Active Ethernet ports with the same SLAN ID.

1 Create the *staged* TLS bridge on an Ethernet port facing the network.

```
zSH> bridge add 1-a-3-0/eth tls vlan 0 slan 200 staged
Adding bridge on 1-a-3-0/eth
Created bridge-interface-record ethernet3-0-200/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
tls          ST    0/200    1/a/3/0/eth      ethernet3-0-200/bridge  UP
1 Bridge Interfaces displayed
```

2 Create the *tagged* TLS bridges facing the subscriber.

```
zSH> bridge add 1-13-1-0/eth tls vlan 0 slan 200 tagged
Adding bridge on 1-13-1-0/eth
Created bridge-interface-record 1-13-1-0-eth-0/bridge
```

```
zSH> bridge add 1-13-2-0/eth tls vlan 0 slan 200 tagged
Adding bridge on 1-13-2-0/eth
Created bridge-interface-record 1-13-2-0-eth-0/bridge
```

```
zSH> bridge add 1-13-3-0/eth tls vlan 0 slan 200 tagged
Adding bridge on 1-13-3-0/eth
Created bridge-interface-record 1-13-3-0-eth-0/bridge
```

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
tls          Tg    0/200    1/13/1/0/eth      1-13-1-0-eth-0/bridge  UP
tls          Tg    0/200    1/13/2/0/eth      1-13-2-0-eth-0/bridge  UP
tls          Tg    0/200    1/13/3/0/eth      1-13-3-0-eth-0/bridge  UP
tls          ST    0/200    1/a/3/0/eth      ethernet3-0-200/bridge  UP
4 Bridge Interfaces displayed
```

Deleting the TLS bridges

Delete the TLS bridges if necessary.

1 Delete the TLS bridge facing the network.

```
zSH> bridge delete ethernet3-0-200/bridge
ethernet3-0-200/bridge Delete complete
```

2 Delete the TLS bridges facing the subscriber.

```
zSH> bridge delete 1-13-1-0-eth-0/bridge
1-13-1-0-eth-0/bridge Delete complete
```

```
zSH> bridge delete 1-13-2-0-eth-0/bridge
1-13-2-0-eth-0/bridge Delete complete
```

```
zSH> bridge delete 1-13-3-0-eth-0/bridge
1-13-3-0-eth-0/bridge Delete complete
```

MXK bridging configuration with VLAN 0 on tagged intralinks

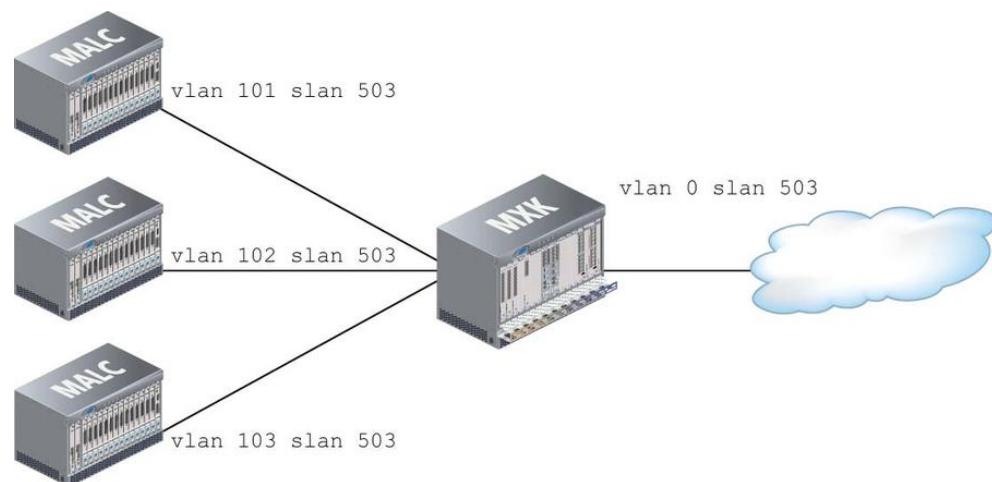
The MXK uses a VLAN wildcard, VLAN ID 0, on double-tagged (*staged*) uplink bridges. This is useful for creating several downstream bridges that have different VLAN IDs but have the same SLAN ID.



Note: Single-tagged VLAN 0 is not allowed.

For example, you might want to subtend several MALCs off of an MXK with different VLAN IDs but the same SLAN ID. In this case, VLAN ID 0 on the uplink bridge will accept all of the VLAN IDs and the same SLAN ID for each subtended MALC. This allows you to configure one uplink bridge that will recognize each of the VLAN IDs and the SLAN ID as shown in [Figure 35](#).

Figure 35: VLAN 0 on the uplink staged



Configuring intralink bridges (tagged to staged configuration)

Creating *tagged* intralink bridges sets the **stripAndInsert** parameter to *false* for the VLAN ID and the **s-tagStripAndInsert** parameter for the SLAN ID to *true*. This causes the strip and insert behavior to strip the SLAN ID on the way to the subtended device and re-insert the SLAN ID on the way to the uplink. The VLAN ID is passed in both directions. The uplink bridge is *staged*, which sets the **stripAndInsert** parameter and the **s-tagStripAndInsert** parameter to *false*. Both the SLAN ID and the VLAN ID are passed on upstream.

- 1 Create *tagged* intralink bridges to the subtended devices.

```

zSH> bridge add 1-13-1-0/eth intralink vlan 101 slan 503 tagged
Adding bridge on 1-13-1-0/eth

```



```
Created bridge-interface-record 1-13-1-0-eth-101/bridge
Bridge-path added successfully
```

```
zSH> bridge add 1-13-2-0/eth intralink vlan 102 slan 503 tagged
Adding bridge on 1-13-2-0/eth
Created bridge-interface-record 1-13-2-0-eth-102/bridge
Bridge-path added successfully
```

```
zSH> bridge add 1-13-3-0/eth intralink vlan 102 slan 503 tagged
Adding bridge on 1-13-3-0/eth
Created bridge-interface-record 1-13-3-0-eth-102/bridge
Bridge-path added successfully
```

2 Verify the bridge and the bridge paths.

```
zSH> bridge show
```

Type	VLAN/SLAN	Orig VLAN/SLAN	Physical	Bridge	St	Table	Data
int		Tg 101/503	1/13/1/0/eth	1-13-1-0-eth-101/bridge	DWN	S	SLAN 503 VLAN 101 Intralink
int		Tg 102/503	1/13/2/0/eth	1-13-2-0-eth-102/bridge	DWN	S	SLAN 503 VLAN 102 Intralink
int		Tg 102/503	1/13/3/0/eth	1-13-3-0-eth-102/bridge	DWN	S	SLAN 503 VLAN 102 Intralink

3 Bridge Interfaces displayed

```
zSH> bridge-path show
```

VLAN/SLAN	Bridge	Address
101/503	1-13-1-0-eth-101/bridge	Intralink
102/503	1-13-2-0-eth-102/bridge	Intralink
103/503	1-13-3-0-eth-103/bridge	Intralink

3 Create the *staged* uplink bridge using VLAN ID 0:

```
zSH> bridge add 1-a-7-0/eth uplink vlan 0 slan 503 staged
Adding bridge on 1-a-7-0/eth
Created bridge-interface-record ethernet7-0-503/bridge
Bridge-path added successfully
```

The uplink bridge accepts any VLAN IDs with the same SLAN ID 503.

Delete bridge

If necessary delete the intralink bridges.

1 Delete the intralink bridges.

```
zSH> bridge delete 1-13-1-0-eth-101/bridge vlan 101
Bridge-path deleted successfully
1-13-1-0-eth-101/bridge delete complete
```

```
zSH> bridge delete 1-13-2-0-eth-102/bridge vlan 102
Bridge-path deleted successfully
1-13-2-0-eth-102/bridge delete complete
```

```
zSH> bridge delete 1-13-3-0-eth-103/bridge vlan 103
Bridge-path deleted successfully
1-13-3-0-eth-103/bridge delete complete
```

2 Delete the uplink bridge:

```
zSH> bridge delete ethernet7-0-503/bridge vlan 0
Bridge-path deleted successfully
ethernet7-0-503/bridge delete complete
```

MXK bridging configuration with VLAN 0 on staged intralinks

In special cases, you can create *staged* intralink bridges from the MXK to subtended MALCs. You do this when there are *untagged* downlink bridges on the MALC to the downstream device, for example, on DSL lines to subscriber phones. In this case, the downstream devices on the MALC do not need the VLAN ID or SLAN ID, but are connected to a network that expects both an SLAN ID and a VLAN ID on the uplink as shown in [Figure 36](#).

Figure 36: Subtended MALCs off the MXK with staged intralinks



Creating the downlink bridge with a VLAN ID and an SLAN ID and using the variable *untagged* causes certain strip and insert behavior. For the *untagged* downlink bridge, both the **stripAndInsert** parameter and the **s-tagstripAndInsert** parameter are set to *true* which causes the VLAN ID and the SLAN ID to be stripped out in the downstream direction, and re-inserted in the upstream direction. Creating an intralink bridge using the variable *staged*, causes both the **stripAndInsert** parameter and the **s-tagstripAndInsert** parameter to be set to false, and both the SLAN ID and the VLAN ID are passed both downstream (to the MALC) and upstream (to the network). This strip and insert behavior on the downlink is called double promotion.



Note: Double promotion, or *untagged* bridges in a network using VLANs and SLANs can only occur on the MALC.

Configuring staged intralink bridge and staged uplink bridge on the MXK and untagged downlink bridge on the MALC

- 1 Create an *staged* uplink bridge and the bridge path for the uplink bridge on the MXK:

```
zSH> bridge add 1-a-4-0/eth uplink vlan 0 slan 503 staged
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-0-503/bridge
Bridge-path added successfully
```

- 2 Create an *staged* intralink bridge on the MXK:

```
zSH> bridge add 1-13-4-0/eth intralink vlan 101 slan 503 staged
Adding bridge on 1-13-4-0/eth
Created bridge-interface-record 1-13-4-0-eth-101-503/bridge
Bridge-path added successfully
```

- 3 Create an *untagged* downlink bridge on the MALC:

```
zSH> bridge add 1-9-1-0/eth downlink vlan 100 slan 500 untagged
Adding bridge on 1-9-1-0/eth
Created bridge-interface-record 1-9-1-0-eth/bridge
```

Bridges with link aggregation

Bridge interfaces can be added to ports that are a part of link aggregation groups.

Configure link aggregation uplink bridges

Creating link aggregated uplink bridges

Unlearned traffic received on this interface is forwarded to the external network.

- 1 To verify link aggregation groups, enter:

```
zSH> linkagg show
LinkAggregations:
slot unit ifName      partner: Sys      Pri    grp ID  status  agg mode
-----
a*  1    1-a-1-0      00:00:00:00:00:00  0x0    0x0    OOS     Active
      links      slot  port  subport      status
-----
      1-a-2-0      a     2     0             ACT
b    1    1-b-1-0      00:00:00:00:00:00  0x0    0x0    OOS     Active
      links      slot  port  subport      status
-----
      1-b-2-0      b     2     0             DSA
global linkagg group red type: red
```

- 2 To create an uplink bridge with link aggregation, enter:

```
zSH> bridge add 1-a-1-0/linkagg uplink vlan 333 tagged
Adding bridge on 1-a-1-0/linkagg
Created bridge-interface-record linkagg-a-1-333/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
upl           Tagged 333    1/a/1/0/linkagg  linkagg-a-1-333/bridge  DWN S VLAN 333 default
1 Bridge Interfaces displayed
```

3 To delete a bridge with link aggregation enter:

```
zSH> bridge delete linkagg-a-1-333/bridge
Bridge-path deleted successfully
linkagg-a-1-333/bridge delete complete
```

Creating an uplink bridge on a aggregated Ethernet port

If a bridge is created on a link aggregated Ethernet interface on a physical port, a linkagg bridge is automatically created.

Create the uplink bridge.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 777
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record linkagg-a-1-777/bridge
Bridge-path added successfully
```

Since the Ethernet port *1-a-2-0/eth* is part of a link aggregation group, the bridge type is automatically designated *linkagg*.

Verify the *linkagg* bridge.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
upl           Tagged 777    1/a/1/0/linkagg  linkagg-a-1-777/bridge  DWN S VLAN 777 default
1 Bridge Interfaces displayed
```

Configure link aggregation line card bridges

Creating a link aggregated bridge on an Ethernet line card

1 Verify the link aggregation group.

```
zSH> linkagg show
LinkAggregations:
slot unit ifName      partner: Sys      Pri    grp ID  status  agg mode
-----
1  1  1-1-1-0      00:00:00:00:00:00  0x0    0x0    OOS    Active
      links      slot  port  subport      status
```

```

-----
1-1-1-0      1      1      0      OOS
1-1-2-0      1      2      0      OOS
global linkagg group red type: red

```

2 Create the bridge on the link aggregation group.

```

zSH> bridge add 1-1-1-0/eth downlink-data vlan 600
Adding bridge on 1-1-1-0/eth
Created bridge-interface-record linkagg-1-1/bridge

```

3 View the bridge created on the link aggregation group.

```

zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
dwn-dat      600    1/1/1/0/linkagg  linkagg-1-1/bridge  DWN
1 Bridge Interfaces displayed

```

Deleting a link aggregation bridge

Delete the link aggregation bridge.

```

zSH> bridge delete linkagg-1-1/bridge
linkagg-1-1/bridge delete complete

```

Configure a TLS bridge on a link aggregation bridge

If a port is a part of a link aggregation group, the bridge type *linkagg* is assigned to the bridge interface.

Configuring a TSL link aggregation bridge on an Ethernet port

In this case, a TLS bridge is created on an uplink card Ethernet port that is a member of a link aggregation group.

- 1 Create the TLS bridge on an Ethernet port. This Ethernet port is a member of a link aggregation group, therefore the bridge interface is assigned *linkagg* as the bridge type.

```

zSH> bridge add 1-a-2-0/eth tls vlan 888
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record linkagg-a-1/bridge
Bridge-path added successfully

```

2 View the TLS bridge.

```

zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
tls      888    1/a/1/0/linkagg  linkagg-a-1/bridge  DWN

```

1 Bridge Interfaces displayed

3 View the TLS bridge path.

```
zSH> bridge-path show
VLAN/SLAN  Bridge                               Address
-----
      888 N/A                               VLAN, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0,
IGMP DSCP: 0, Flap Mode: Fast
```

The bridge-path on TLS bridges are on the VLAN ID, not the bridge interface and are created only for the first instance of TLS and VLAN ID.

Configuring a TLS link aggregation bridge on a link aggregation group

In this case, a TLS bridge is created on a link aggregation group.

1 Verify the linkagg group.

```
zSH> linkagg show
LinkAggregations:
slot unit ifName      partner: Sys      Pri      grp ID  status  agg mode
-----
a*  1  1-a-1-0      00:00:00:00:00:00  0x0      0x0      OOS      Active
    links      slot  port  subport      status
-----
    1-a-2-0      a      2      0              ACT
b   1  1-b-1-0      00:00:00:00:00:00  0x0      0x0      OOS      Active
    links      slot  port  subport      status
-----
    1-b-2-0      b      2      0              DSA
global linkagg group red type: red
```

2 Create a TLS bridge on the linkagg group interface.

```
zSH> bridge add 1-a-1-0/linkagg tls vlan 888
Adding bridge on 1-a-1-0/linkagg
Created bridge-interface-record linkagg-a-1/bridge
Bridge-path added successfully
```

The bridge-path on TLS bridges are on the VLAN ID, not the bridge interface and are created only for the first instance of TLS and VLAN ID.

3 Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
tls      888      1/a/1/0/linkagg  linkagg-a-1/bridge  DWN
1 Bridge Interfaces displayed
```

4 View the TLS bridge-path.

```
zSH> bridge-path show
VLAN/SLAN  Bridge                               Address
```

888 N/A
IGMP DSCP: 0, Flap Mode: Fast

VLAN, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0,

Bridge loop prevention

This section covers:

- [Bridge loop prevention overview, page 276](#)
- [Configure bridge loop prevention, page 277](#)
- [View bridge loop detection alarms, page 280](#)
- [View bridge loop prevention on a bridge, page 281](#)
- [Unblock the bridge, page 281](#)

Bridge loop prevention overview

This section covers:

- [Bridge loop prevention on asymmetrical bridges, page 276](#)
- [Bridge loop prevention on TLS bridges, page 276](#)

Bridge loop prevention can be configured on either asymmetrical or TLS bridges to resolve certain incorrect MAC address behaviors.

Bridge loop prevention on asymmetrical bridges

Bridge loop prevention can be configured on the bridge path of the bridge interface when a MAC address on asymmetrical bridges is seen as coming in on both the uplink and the downlink.

When bridge loop behavior occurs and *block blockAsym* is configured on the uplink bridge interface with VLAN ID the system blocks the downlink after detecting this incorrect MAC address behavior.

After the blocked bridge receives an offending MAC address, the system sends a MAJOR alarm that indicates a bridge was blocked to prevent a loop. This alarm displays the bridge interface and the offending MAC address.

In this case, the blocked bridge interface must be unblocked with the **bridge unblock interface/type** command.

When bridge loop behavior occurs and *block blockAsymAuto* is configured on the uplink bridge interface with VLAN ID, the system initiates a series of three cyclic monitoring checks to see if the bridge loop condition is resolved. If the bridge loop condition is resolved, the bridge interface is automatically unblocked and a bridge loop clear alarm is sent.

If the condition is not resolved, the MAJOR alarm is cleared and a CRITICAL alarm is sent. In this case, the blocked bridge interface must be unblocked with the **bridge unblock interface/type** command.

Bridge loop prevention on TLS bridges

Bridge loop prevention can be configured on the bridge path of a TLS bridge when a MAC address is seen as coming in on one TLS bridge and then as coming in on another TLS bridge.

When this behavior occurs and *block blockall* is configured on the VLAN ID of the TLS bridges, the system blocks the second TLS bridge and then sends a MAJOR alarm describing the second TLS bridge that saw the MAC address. The bridge is then blocked to prevent a loop.

In this case, the blocked bridge interface must be unblocked with the **bridge unblock interface/type** command.

When bridge loop behavior occurs and *block blockAsymAuto* is configured on the TLS bridge interface with VLAN ID, the system initiates a series of three cyclic monitoring checks to see if the bridge loop condition is resolved. If the

bridge loop condition is resolved, the bridge interface is automatically unblocked and a bridge loop clear alarm is sent.

If the condition is not resolved, the MAJOR alarm is cleared and a CRITICAL alarm is sent. In this case, the blocked bridge interface must be unblocked with the **bridge unblock interface/type** command.

Configure bridge loop prevention

Configuring bridge loop prevention on asymmetric bridges with blockAsym

- 1 Create the asymmetrical bridging configuration.

Create an uplink bridge.

```
zSH> bridge add 1-a-4-0/eth uplink vlan 100
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-100/bridge
Bridge-path added successfully
```

- 2 Modify the bridge path to enable asymmetrical bridge blocking using **bridge-path modify interface/type vlan default block blockasym**.

```
zSH> bridge-path modify ethernet4-100/bridge vlan 100 default block
blockAsym
Bridge-path ethernet4-100/bridge/3/100/0/0/0/0/0/0/0 has been modified
```



Note: Enter exactly the same command syntax to enable blocking on an existing bridge path. The existing bridge path will be overwritten, and blocking will be enabled.

View the bridge path.

```
zSH> bridge-path show
VLAN/SLAN  Bridge                                     Address
-----
100 ethernet4-100/bridge                 Default, Age: 3600, MCAST Age: 250, IGMP Query Interval:
0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym
```

- 3 Create a downlink bridge.

```
zSH> bridge add 1-6-1-501/gponport gtp 1 downlink-data vlan 100 tagged
Adding bridge on 1-6-1-501/gponport
Created bridge-interface-record 1-6-1-501-gponport-100/bridge
```

View the bridges.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
dwn-dat  Tagged 100  1/6/1/1/gpononu  1-6-1-501-gponport-100/bridge  DWN
upl      Tagged 100  1/a/4/0/eth      ethernet4-100/bridge           DWN S VLAN 100 default
```

2 Bridge Interfaces displayed

Configuring bridge loop prevention on asymmetric bridges with blockAsymAuto

- 1 Create the asymmetrical bridging configuration.

Create an uplink bridge.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 200 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-200/bridge
Bridge-path added successfully
```

- 2 Modify the bridge path to enable asymmetrical bridge auto unblocking using **bridge-path modify interface/type vlan default block blockAsymAuto**.

```
zSH> bridge-path modify ethernet2-200/bridge vlan 200 default block blockAsymAuto
Bridge-path ethernet2-200/bridge/3/200/0/0/0/0/0/0/0 has been modified
```

View the bridge bath:

```
zSH> bridge-path show
VLAN/SLAN  Bridge                                     Address
-----
200 ethernet2-200/bridge      Default, Age: 3600, MCAST Age: 250, IGMP Query Interval:
0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym/Auto
```

- 3 Create a downlink bridge on the same VLAN ID.

```
zSH> bridge add 1-1-6-0/eth downlink-data vlan 200 tagged
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth-200/bridge
```

View the bridges:

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
dwn-dat      Tagged 200  1/1/6/0/eth  1-1-6-0-eth-200/bridge  DWN
upl          Tagged 200  1/a/2/0/eth  ethernet2-200/bridge    UP  S VLAN 200 default
2 Bridge Interfaces displayed
```

Configuring bridge loop prevention on TLS bridges with blockAll

- 1 Create the network facing TLS bridge.

```
zSH> bridge add 1-a-4-0/eth tls vlan 999
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4/bridge
Bridge-path added successfully
```

- 2 Modify the bridge path on the VLAN ID to enable TLS bridge blocking using **bridge-path modify interface/type vlan <vlanid> block blockasym**.

```
zSH> bridge-path modify vlan 999 block blockAll
Bridge-path /14/999/0/0/0/0/0/0/0 has been modified
```

- 3 View the bridge-path.

```
zSH> bridge-path show
VLAN/SLAN   Bridge                               Address
-----
          999 N/A                               VLAN, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0,
IGMP DSCP: 0, Flap Mode: Fast, Block: All
```

- 4 Create the subscriber facing TLS bridges.

```
zSH> bridge add 1-6-12-0/eth tls vlan 999
Adding bridge on 1-6-12-0/eth
Created bridge-interface-record 1-6-12-0-eth/bridge
```

```
zSH> bridge add 1-6-13-0/eth tls vlan 999
Adding bridge on 1-6-13-0/eth
Created bridge-interface-record 1-6-13-0-eth/bridge
```

Verify the bridges.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical      Bridge          St  Table Data
-----
tls           999       1/6/12/0/eth  1-6-12-0-eth/bridge  DWN
tls           999       1/6/13/0/eth  1-6-13-0-eth/bridge  DWN
tls           999       1/a/4/0/eth   ethernet4/bridge     DWN
3 Bridge Interfaces displayed
```

Configuring bridge loop prevention on TLS bridges with blockAllAuto

- 1 Create the network facing TLS bridge.

```
zSH> bridge add 1-a-3-0/eth tls vlan 700
Adding bridge on 1-a-3-0/eth
Created bridge-interface-record ethernet3/bridge
Bridge-path added successfully
```

- 2 Modify the bridge path on the VLAN ID to enable TLS bridge blocking using **bridge-path modify interface/type vlan <vlanid> block blockasym**.

```
zSH> bridge-path modify vlan 700 block blockAllAuto
Bridge-path /14/700/0/0/0/0/0/0/0 has been modified
```

- 3 View the bridge-path.

```
zSH> bridge-path show
```

VLAN/SLAN	Bridge	Address

700 N/A	VLAN, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0, IGMP DSCP: 0, Flap Mode: Fast, Block: All/Auto	

4 Create the subscriber facing TLS bridges.

```
zSH> bridge add 1-6-1-0/eth tls vlan 700
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth/bridge

zSH> bridge add 1-6-2-0/eth tls vlan 700
Adding bridge on 1-6-2-0/eth
Created bridge-interface-record 1-6-2-0-eth/bridge
```

Verify the bridges.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
tls           700      1/6/1/0/eth  1-6-1-0-eth/bridge  UP
tls           700      1/6/2/0/eth  1-6-2-0-eth/bridge  DWN
tls           700      1/a/3/0/eth  ethernet3/bridge    DWN
3 Bridge Interfaces displayed
```

View bridge loop detection alarms

Viewing loop detected alarms

- 1 On the console, the following alarm appears when a loop is detected.

```
zSH> JUN 22 02:12:40: alert : 1/a/1093: bridge: BridgeTrapSend(): l=1223:
tBridgeMain: Bridge Loop detected on 1-10-1-501-gponport-100:(0/100/
00:15:C5:3A:A3:B8) .
```

- 2 Enter **alarm show** to display the loop detection alarm at the system level.

```
zSH> alarm show
***** Central Alarm Manager *****
ActiveAlarmCurrentCount :13
AlarmTotalCount :16
ClearAlarmTotalCount :3
OverflowAlarmTableCount :0
ResourceId AlarmType AlarmSeverity
-----
1-a-2-0/eth linkDown critical
1-a-3-0/eth linkDown critical
1-a-6-0/eth linkDown critical
1-a-7-0/eth linkDown critical
1-a-8-0/eth linkDown critical
1-a-9-0/eth linkDown critical
1-a-10-0/eth linkDown critical
1-a-11-0/eth linkDown critical
1-10-2-0/gponolt linkDown critical
```

```

1-10-3-0/gponolt linkDown critical
1-10-4-0/gponolt linkDown critical
system not_in_redundant_mode major
1-10-1-501-gponport-100 bridgeLoopDetect 0/100/00:15:C5:3A:A3:B8 major

```

View bridge loop prevention on a bridge

All bridges that are blocked by bridge loop protection, RSTP, or EAPS are displayed with the **bridge show blk** command.



Note: The **bridge show blk** command displays bridges that are normally blocked in EAPS or RSTP configurations.

Bridges configured with the *block blockassym* variable for bridge loop prevention will display the MAC address as well as the bridge interface name. Bridges blocked as a normal part of RSTP or EAPS configurations do not display MAC addresses and should remain blocked. Do not unblock the RSTP and EAPS interfaces.

Finding bridges that were blocked by bridge loop protection

Enter the **bridge show blk** command to view blocked bridges.

This example confirms that there are no existing blocked bridges.

```

zSH> bridge show blk
No Bridge Interfaces found.

```

This example confirms that a blocked bridge exists.

A bridge loop alarm appears in the console window.

```

zSH> AUG 05 19:38:38: alert   : 1/b/1062: bridge: BridgeTrapSend():
l=1233: tBridgeMain: Bridge Loop detected on
1-9-4-0-eth-100:(0/100/00:00:00:00:00:04) .
AUG 05 19:38:42: alert   : 1/a/1093: bridge: BridgeTrapSend(): l=1233:
tBridgeMain: Bridge Loop detected on
1-9-4-0-eth-100:(0/100/00:00:00:00:00:04) .

```

The **bridge show blk** command displays a blocked bridge.

```

zSH> bridge show blk
      Orig
Type VLAN/SLAN  VLAN/SLAN  Physical
Bridge          St   Table Data
-----
dwn             Tagged 100   1/9/4/0/eth          1-9-4-0-eth-100/bridge  BLK  A 00:00:00:00:00:04
1 Bridge Interfaces displayed

```

Unblock the bridge

The syntax for unblocking a blocked bridge is:

```

bridge unblock <interface>/<type> |
      [slot <slotNum>]

```

```

[vlan <vlanId>]
[slan <slanId>]
[vlan-count <value>]
[mvr [<mvrVlan>]]
[secure]
[uplink | downlink | intralink | tls | rlink | pppoa | wire |
 mvr | user | downlink-video | downlink-data | downlink-pppoe |
 downlink-p2p | downlink-voice | downlink-upmcast | ipob-tls |
 ipob-uplink | ipob-downlink]
[verbose]

```

Unblocks bridge interfaces which have been blocked due to bridge storm detection (BSD) and due to bridge loop detection.

Where:

```

<interface>/<type>
    The interface can be a bridge, GPON OLT, Ethernet Port, etc.
    Wildcard formats are supported. The interface must come immediately
    after "bridge unblock".
slot <slotNum>
    Process all bridge interfaces for ports in the specified slot.
    <slotNum> may be a single number, a bracketed list containing
    comma-separated numbers or a dash-separated number range or a
    combination of both.
vlan <vlanId>
    Process all bridge interfaces for the specified VLAN.
    <vlanId> may be a single number, a bracketed list containing
    comma-separated numbers or a dash-separated number range or a
    combination of both.
vlan-count <count>
    Process bridges that have VLAN ID values in the range
    <vlan> to <vlan+count>
slan <slanId>
    Process all bridge interfaces for the specified SLAN.
    <slanId> may be a single number, a bracketed list containing
    comma-separated numbers or a dash-separated number range or a
    combination of both.
secure
    Process secure bridges.
mvr [<mvrVlan>]
    Process all bridge interfaces associated with the given MVR
    vlan. <mvrVlan> may be a single number, a bracketed list containing
    comma-separated numbers or a dash-separated number range or a
    combination of both. If no MVR vlan or 0 is entered, all MVR related
    bridges are processed.
uplink | downlink | intralink | tls | rlink | pppoa | wire |
mvr | user | downlink-video | downlink-data | downlink-pppoe |
downlink-p2p | downlink-voice | downlink-upmcast | ipob-tls |
ipob-uplink | ipob-downlink]
    Process bridges of the specified bridge-type. Multiple bridge
    types can be specified.
verbose
    display "unblock" operation status

```

Unblocking the bridge

For example, to unblock a bridge that is blocked because of loop prevention using the bridge interface enter.

```
zSH> bridge unblock 1-10-1-501-gponport/bridge
```

The following type of information is displayed in the console window.

```
zSH> JUN 22 02:14:15: alert : 1/a/1027: bridge: BridgeTrapSend(): l=1233:
tCliInit0: Bridge Loop Alarm for 1-10-1-501-gponport-100 cleared.
```

To unblock a bridge using the slot number and VLAN ID enter:

```
zSH> bridge unblock slot 5 vlan 100
```

To unblock a bridge using the VLAN ID enter:

```
zSH> bridge unblock vlan 100
```

Secure bridging

This section describes dynamic IP filtering on a bridge (secure DHCP) and how to configure static IP and static MAC for secure bridging:

- [Dynamic IP filtering on a bridge \(Secure DHCP\), page 283](#)
- [Static IP and MAC for secure bridging on the MXK, page 284](#)

Dynamic IP filtering on a bridge (Secure DHCP)



Note: MXK uplinks and *network* facing TLS bridges should NOT be configured with a *secure* filter because there are no DHCP client responses possible from network facing bridges. If *secure* is configured on uplink or TLS network facing bridges, traffic will not pass.



Note: For GPON ports, adding *secure* to one VLAN ID will secure the entire port and *all* bridges configured on that port with the same VLAN ID.

The MXK enables secure DHCP settings on downlink bridges, *subscriber* facing TLS bridges, and GPON ports to prevent a user with a statically configured IP address from bypassing DHCP security enforcement. This filter blocks users from accessing the network using anything other than the valid DHCP offered IP address.

When packets are received or sent out a secure downlink bridge interface, TLS subscriber facing bridge interface, or GPON port and VLAN, the MXK checks the IP address against the dynamic IP bridge filter. If a match is found (the address was provided by the DHCP server), the packet is allowed to pass through the filter. Otherwise, it is blocked.

The unicast aging setting for allowed packets is determined based on the DHCP lease time.

Configuring a dynamic IP filter on a bridge

A dynamic IP filter can be configured, modified, and deleted using the **bridge add**, **modify**, or **delete** commands.

Create a downlink bridge using the **bridge add** command with the **secure** option to create the dynamic IP filter. The **secure** option creates two static bridge paths (MAC and IP) for each host on the bridge that successfully negotiates its IP address from the DHCP server.

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 109 slan 509 tagged secure
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-109/bridge
```

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
dwn-dat      Tg 109/509  1/6/1/0/eth      1-6-1-0-eth-109/bridge  UP
1 Bridge Interfaces displayed
```

Deleting the dynamic IP filter on a bridge

Delete the dynamic IP on a bridge filter if necessary.

```
zSH> bridge delete 1-6-1-0-eth-109/bridge
1-6-1-0-eth-109/bridge delete complete
```

Static IP and MAC for secure bridging on the MXK

This section describes secure bridging on downlink and subscriber facing TLS bridges and includes:

- [Configure static mac and IP on downlink bridges, page 285](#)
 - [Case 1: Configuring a secure downlink bridge with static mac+ip, page 285](#)
 - [Case 2: Configuring a secure downlink bridge with static MAC, page 286](#)
 - [Case 3: Configuring a secure downlink bridge with static ip, page 287](#)
- [Configure static mac and IP on TLS bridges, page 288](#)
 - [Case 4: Configuring a secure subscriber facing TLS bridge with static mac+ip, page 288](#)
 - [Case 5: Configuring a secure subscriber facing TLS bridge with static mac address, page 290](#)
 - [Case 6: Configuring a secure TLS bridge with static ip, page 291](#)

The MXK allows secure bridge settings on downlink bridges and *subscriber* facing TLS bridges that will only accept traffic for the configured MAC and/or IP addresses. Secure static bridging prevents users from accessing the network by using any MAC or IP address other than the one that is configured.

When packets are received or sent out a secure downlink bridge interface or TLS subscriber facing bridge interface, the MXK checks the IP or MAC address against the configured IP or MAC address and if a match is found the packet is sent on to the network. If the packet does not match, the packet is discarded.

Configure static mac and IP on downlink bridges

Case 1: Configuring a secure downlink bridge with static mac+ip

In this case both the MAC address and the IP are statically configured on a secure downlink bridge.

- 1 Create the secure downlink bridge using the keywords *secure*, *static*, and *mac+ip*.

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 222 tagged secure static mac+ip
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth/bridge
```

- 2 Verify the bridge.

```
zSH> bridge show
      Orig
Type   VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
dwn-dat      Tagged 222  1/6/1/0/eth  1-6-1-0-eth-222/bridge  UP
1 Bridge Interfaces displayed
```

- 3 Configure two bridge paths with the **bridge-path add** command to add the static MAC address and then the static IP address to the secure downlink bridge.

```
zSH> bridge-path add 1-6-1-0-eth/bridge vlan 222 mac 00:0B:BD:14:B0:26
Bridge-path added successfully
```

```
zSH> bridge-path add 1-6-1-0-eth/bridge vlan 222 ip 10.11.12.111
Bridge-path added successfully
```



Note: For IPv6 compatibility use the **ipv6** keyword in the **bridge-path add/modify** command.

- 4 View the secure downlink bridge now configured with a static MAC address and a static IP address.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
dwn-dat      Tagged 222  1/6/1/0/eth  1-6-1-0-eth-222/bridge  UP  S 00:0b:bd:14:b0:26
                                   S 10.11.12.111

1 Bridge Interfaces displayed
```

- 5 Verify the static MAC and IP addresses configured on the bridge path.

```
zSH> bridge-path show
VLAN/SLAN  Bridge  Address
-----
222 1-6-1-0-eth/bridge  00:0b:bd:14:b0:26
222 1-6-1-0-eth/bridge  10.11.12.111
```

Deleting the secure downlink bridge with static mac+ip

- 1 Delete the two bridge paths with the static MAC address and the static IP address *before* deleting the secure downlink bridge.

```
zSH> bridge-path delete 1-6-1-0-eth-222/bridge vlan 222 ip 10.11.12.111
Delete complete
```



Note: For IPv6 compatibility use the **ipv6** keyword in the **bridge-path add/modify** command.

```
zSH> bridge-path delete 1-6-1-0-eth-222/bridge vlan 222 mac 00:0B:BD:14:B0:26
Delete complete
```

- 2 Delete the secure downlink bridge.

```
zSH> bridge delete 1-6-1-0-eth-222/bridge vlan 222
1-6-1-0-eth-222/bridge delete complete
```

Case 2: Configuring a secure downlink bridge with static MAC

In this case the MAC address is statically configured on a secure downlink bridge.

- 1 Create a secure downlink bridge using the keywords *secure*, *static*, and *mac*.

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 200 tagged secure static mac
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-200/bridge
```

- 2 Verify the bridge.

```
zSH> bridge show
      Orig
```

Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-dat		Tagged 200	1/6/1/0/eth	1-6-1-0-eth-200/bridge	UP	

1 Bridge Interfaces displayed

- 3 Configure a bridge path with the **bridge-path add** command to add the static MAC address to the secure downlink bridge.

```
zSH> bridge-path add 1-6-1-0-eth-200/bridge vlan 200 00:0B:BD:14:B0:26
Bridge-path added successfully
```

- 4 View the secure downlink bridge now configured with a static MAC address.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
dwn-dat   Tagged 200  1/6/1/0/eth  1-6-1-0-eth-200/bridge  UP  S 00:0b:bd:14:b0:26
1 Bridge Interfaces displayed
```

- 5 View the bridge path.

```
zSH> bridge-path show
VLAN/SLAN  Bridge      Address
-----
200 1-6-1-0-eth-200/bridge  00:0b:bd:14:b0:26
```

Deleting the secure downlink bridge with static MAC

- 1 Delete the bridge path with the MAC address *before* deleting the secure downlink bridge.

```
zSH> bridge-path delete 1-6-1-0-et-200/bridge vlan 200 mac 00:0b:db:14:b0:26
Delete complete
```

- 2 Delete the secure downlink bridge.

```
zSH> bridge delete 1-6-1-0-eth-200/bridge vlan 200
1-6-1-0-eth-200/bridge delete complete
```

Case 3: Configuring a secure downlink bridge with static ip

In this case the IP is statically configured on a secure downlink bridge.

- 1 Create the secure downlink bridge using the keywords *secure*, *static*, and *ip*.

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 300 secure static ip
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-300/bridge
```

- 2 Verify the bridge.

```
zSH> bridge show
Orig
```

Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-dat		Tagged 300	1/6/1/0/eth	1-6-1-0-eth-300/bridge	UP	

1 Bridge Interfaces displayed

- 3 Configure a bridge path with the **bridge-path add** command to add the static IP address to the secure downlink bridge.

```
zSH> bridge-path add 1-6-1-0-eth-300/bridge vlan 300 ip 10.11.12.111
Bridge-path added successfully
```



Note: For IPv6 compatibility use the **ipv6** keyword in the **bridge-path add/modify** command.

- 4 View the secure downlink bridge now configured with a static IP address.

```
zSH> bridge show
```

Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-dat		Tagged 300	1/6/1/0/eth	1-6-1-0-eth-300/bridge	UP	S 10.11.12.111

1 Bridge Interfaces displayed

- 5 View the bridge path.

```
zSH> bridge-path show
```

VLAN/SLAN	Bridge	Address
300	1-6-1-0-eth-300/bridge	10.11.12.111

Deleting the secure downlink bridge with static ip

- 1 Delete the bridge path with the MAC address before deleting the secure downlink bridge.

```
zSH> bridge-path delete 1-6-1-0-eth-300/bridge vlan 300 ip 10.11.12.111
Delete complete
```



Note: For IPv6 compatibility use the **ipv6** keyword in the **bridge-path add/modify** command.

- 2 Delete the secure downlink bridge.

```
zSH> bridge delete 1-6-1-0-eth-300/bridge vlan 300
1-6-1-0-eth-300/bridge delete complete
```

Configure static mac and IP on TLS bridges

Case 4: Configuring a secure subscriber facing TLS bridge with static mac+ip

In this case, both the MAC address and the IP are statically configured on a secure *tls* bridge

- 1 Create the secure subscriber facing TLS bridge using the keywords *secure*, *static*, and *mac+ip*.

```
zSH> bridge add 1-6-1-0/eth tls vlan 200 secure static mac+ip
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth/bridge
Bridge-path added successfully
```

- 2 Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
tls             200        1/6/1/0/eth      1-6-1-0-eth/bridge  UP
1 Bridge Interfaces displayed
```

For TLS bridges, the first time a TLS bridge is created with a VLAN, a bridge path is automatically created on the VLAN. Since this bridge path is created on the VLAN, additional bridge paths must be configured on the bridge interface to associate the secure MAC address and the secure IP address to the TLS bridge.

```
zSH> bridge-path show
VLAN/SLAN  Bridge          Address
-----
200 N/A      VLAN, Age: 3600, MCAST Age: 250, IGMP Query
Interval: 0, IGMP DSCP: 0, Flap Mode: Fast
```

- 3 Configure two bridge paths with the **bridge-path add** command to add the static MAC address and the static IP address to the secure TLS bridge.

```
zSH> bridge-path add 1-6-1-0-eth/bridge vlan 200 mac 00:0B:BD:14:B0:26
Bridge-path added successfully
```

```
zSH> bridge-path add 1-6-1-0-eth/bridge vlan 200 ip 10.11.12.111
Bridge-path added successfully
```



Note: For IPv6 compatibility use the **ipv6** keyword in the **bridge-path add/modify** command.

- 4 View the secure TLS bridge now configured with a static MAC address and a static IP address.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
tls             200        1/6/1/0/eth      1-6-1-0-eth/bridge  UP  S 00:0b:bd:14:b0:26
                                   S 10.11.12.111
1 Bridge Interfaces displayed
```

- 5 Verify the static MAC and IP addresses configured on the bridge path.

```
zSH> bridge-path show
```

VLAN/SLAN	Bridge	Address
200	N/A	VLAN, Age: 3600, MCAST Age: 250, IGMP Query
Interval: 0, IGMP DSCP: 0, Flap Mode: Fast		
200	1-6-1-0-eth/bridge	00:0b:bd:14:b0:26
200	1-6-1-0-eth/bridge	10.11.12.111

Deleting the secure TLS bridge with static mac+ip

- 1 Delete the two bridge paths with the MAC address and the IP address before deleting the *tls* bridge.

```
zSH> bridge-path delete 1-6-1-0-eth/bridge vlan 200 ip 10.11.12.111
Delete complete
```

```
zSH> bridge-path delete 1-6-1-0-eth/bridge vlan 200 mac 00:0b:db:14:b0:26
Delete complete
```

- 2 Delete the secure TLS bridge.

```
zSH> bridge delete 1-6-1-0-eth/bridge
Bridge-path deleted successfully
1-6-1-0-eth/bridge delete complete
```

Case 5: Configuring a secure subscriber facing TLS bridge with static mac address

In this case a MAC address is statically configured on a secure subscriber facing TLS bridge.

- 1 Create a secure *tls* bridge using the keywords *secure*, *static*, and *mac*.

```
zSH> bridge add 1-6-1-0/eth tls vlan 200 secure static mac
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth/bridge
Bridge-path added successfully
```

- 2 Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
tls           200        1/6/1/0/eth  1-6-1-0-eth/bridge  UP
1 Bridge Interfaces displayed
```

For TLS bridges, the first time a TLS bridge is created with a VLAN, a bridge path is automatically created on the VLAN. Since this bridge path is created on the VLAN, an additional bridge path must be configured on the bridge interface to associate the secure MAC address to the TLS bridge.

```
zSH> bridge-path show
VLAN/SLAN  Bridge      Address
-----
```

```

200 N/A
Interval: 0, IGMP DSCP: 0, Flap Mode: Fast
VLAN, Age: 3600, MCAST Age: 250, IGMP Query

```

- 3 Configure a bridge path with the **bridge-path add** command to add the static MAC address to the secure *tls* bridge.

```

zSH> bridge-path add 1-1-6-0-eth/bridge vlan 200 mac 00:0B:BD:14:B0:26
Bridge-path added successfully

```

- 4 View the secure *tls* bridge now configured with a static MAC address.

```

zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
tls             200        1/6/1/0/eth      1-6-1-0-eth/bridge  UP  S 00:0b:bd:14:b0:26
1 Bridge Interfaces displayed

```

- 5 View the bridge path with the MAC address.

```

zSH> bridge-path show
VLAN/SLAN  Bridge          Address
-----
200 N/A
Interval: 0, IGMP DSCP: 0, Flap Mode: Fast
200 1-6-1-0-eth/bridge 00:0b:bd:14:b0:26
VLAN, Age: 3600, MCAST Age: 250, IGMP Query

```

Deleting the secure TLS bridge with static mac

- 1 Delete the bridge paths with the static MAC address before deleting the secure *tls* bridge

```

zSH> bridge-path delete 1-6-1-0-eth/bridge vlan 200 mac 00:0b:bd:14:b0:26
Delete complete

```

- 2 Delete the secure downlink bridge.

```

zSH> bridge delete 1-6-1-0-eth/bridge vlan 200
Bridge-path deleted successfully
1-6-1-0-eth/bridge delete complete

```

Case 6: Configuring a secure TLS bridge with static ip

In this case the IP is statically configured on a secure *tls* bridge.

- 1 Create the secure *tls* bridge using the keywords *secure*, *static*, and *ip*.

```

zSH> bridge add 1-6-1-0/eth tls vlan 200 secure static ip
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth/bridge
Bridge-path added successfully

```

- 2 Verify the bridge.

```

zSH> bridge show
Orig

```

Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
tls		200	1/6/1/0/eth	1-6-1-0-eth/bridge	UP		
1 Bridge Interfaces displayed							

For TLS bridges, the first time a TLS bridge is created with a VLAN, a bridge path is automatically created on the VLAN. Since this bridge path is created on the VLAN, an additional bridge path must be configured on the bridge interface to associate the secure IP address to the TLS bridge.

```
zSH> bridge-path show
VLAN/SLAN      Bridge                               Address
-----
200 N/A                               VLAN, Age: 3600, MCAST Age: 250, IGMP Query
Interval: 0, IGMP DSCP: 0, Flap Mode: Fast
```

- 3 Configure a bridge path with the **bridge-path add** command to add the static IP address to the secure *tls* bridge.

```
zSH> bridge-path add 1-6-1-0-eth/bridge vlan 200 ip 10.11.12.111
Bridge-path added successfully
```



Note: For IPv6 compatibility use the **ipv6** keyword in the **bridge-path add/modify** command.

- 4 View the secure *tls* bridge now configured with a static IP address.

```
zSH> bridge show
```

Orig							
Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
tls		200	1/6/1/0/eth	1-6-1-0-eth/bridge	UP	S	10.11.12.111
1 Bridge Interfaces displayed							

- 5 View the bridge path with the IP address.

```
zSH> bridge-path show
VLAN/SLAN      Bridge                               Address
-----
200 N/A                               VLAN, Age: 3600, MCAST Age: 250, IGMP Query
Interval: 0, IGMP DSCP: 0, Flap Mode: Fast
200 1-6-1-0-eth/bridge                10.11.12.111
```

Deleting the secure TLS bridge with static ip

- 1 Delete the bridge path with the MAC address before deleting the secure *tls* bridge.

```
zSH> bridge-path delete 1-6-1-0-eth/bridge vlan 200 ip 10.11.12.111
Delete complete
```

- 2 Delete the secure *tls* bridge.

```
zSH> bridge delete 1-6-1-0-eth/bridge vlan 200
Bridge-path deleted successfully
```


1-6-1-0-eth/bridge delete complete

Broadcast suppression

Broadcast suppression enables DHCP information to be relayed between DHCP client and host while broadcast filtering is enabled.

The **bridgeIfCustomDHCP** setting enables bridge interfaces to pass DHCP information independent of the filterBroadcast setting. Setting **bridgeIfCustomDHCP** to *true* will cause that bridge interface to pass DHCP OFFER and ACK packets even though the filterBroadcast is set to *true*.

To enable bridgeIfCustomDHCP on an existing bridge, update the **bridge-interface-record**.

```
zSH> update bridge-interface-record 1-13-1-0-eth-101/bridge
bridge-interface-record 1-13-1-0-eth-101/bridge
Please provide the following: [q]uit.
vpi: -----> {0}:
vci: -----> {0}:
vlanId: -----> {101}:
stripAndInsert: -----> {false}:
customARP: -----> {false}:
filterBroadcast: -----> {false}:
learnIp: -----> {false}:
learnUnicast: -----> {false}:
maxUnicast: -----> {0}:
learnMulticast: -----> {false}:
forwardToUnicast: -----> {false}:
forwardToMulticast: -----> {false}:
forwardToDefault: -----> {true}:
bridgeIfCustomDHCP: -----> {false}: true
bridgeIfIngressPacketRuleGroupIndex: -> {0}:
vlanIdCOS: -----> {0}:
outgoingCOSOption: -----> {disable}:
outgoingCOSValue: -----> {0}:
s-tagTPID: -----> {0x8100}:
s-tagId: -----> {501}:
s-tagStripAndInsert: -----> {true}:
s-tagOutgoingCOSOption: -----> {s-tagdisable}:
s-tagIdCOS: -----> {0}:
s-tagOutgoingCOSValue: -----> {0}:
mcastControlList: -----> {}:
maxVideoStreams: -----> {0}:
isPPPoA: -----> {false}:
floodUnknown: -----> {false}:
floodMulticast: -----> {false}:
bridgeIfEgressPacketRuleGroupIndex: --> {0}:
bridgeIfTableBasedFilter: -----> {NONE(0)}:
bridgeIfDhcpLearn: -----> {NONE(0)}:
mvrVlan: -----> {0}
vlan-xlate-from: -----> {0}
slan-xlate-from: -----> {0}
bridge-type: -----> {downlink}:
```

```
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Configure uplink and downlink bridges on GPON for triple-play services



Note: All bridges on GPON ports must have a VLAN ID and must be designated *tagged*. GPON does not support *untagged* bridging.



Note: For information on Smart OMCI and ONU management, see [Chapter 10, MXK GPON Cards](#). For more information on configuring bridged video on the MXK, see [Chapter 5, Video Configuration](#).

You can create bridges on GEM ports to provide triple-play services. Bridges must be created to pass traffic between the MXK and the upstream data, voice, and video source, and the downstream ONUs.

You create the GEM port with **bridge add**. For different services, you can associate different GPON traffic profiles with different GEM ports.



Note: If an ONU is activated with Smart OMCI, when you use **bridge add** to create a GEM port, be sure that the GEM port ID matches the GEM index specified in the Smart OMCI web-interface.

Configuring an uplink bridge and downlink bridge on a GEM port for data services

Create an uplink and downlink bridge on a GEM port for data services.

- 1 Create the tagged uplink bridge with a VLAN ID.

```
zSH> bridge add 1-a-4-0/eth uplink vlan 100 tagged
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-100/bridge
Bridge-path added successfully
```

- 2 Create the GPON traffic profile for the downlink bridge for data services.

```
zSH> new gpon-traffic-profile 1
gpon-traffic-profile 1
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}: 512
traffic-class: -----> {ubr}:
compensated: -----> {false}:
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: ----> {0}:
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: -----> {0}:
```

```
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

3 Create the downlink bridge with the GPON traffic profile and VLAN 100.

```
zSH> bridge add 1-6-1-501/gponport gtp 1 downlink-data vlan 100 tagged
Adding bridge on 1-6-1-501/gponport
Created bridge-interface-record 1-6-1-501-gponport-100/bridge
```

Configuring an uplink bridge and downlink bridge on a GEM port for voice services

Create an uplink and downlink bridge on a GEM port for voice services.

1 Create the tagged uplink bridge with a VLAN ID.

```
zSH> bridge add 1-a-4-0/eth uplink vlan 200 tagged
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-200/bridge
```

2 Create the GPON traffic profile for the downlink bridge for voice services.

```
zSH> new gpon-traffic-profile 2
gpon-traffic-profile 2
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}: 512
traffic-class: -----> {ubr}: cbr
compensated: -----> {false}: true
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: ----> {0}:
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: ----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

3 Create the downlink bridge with the GPON traffic profile and VLAN 200.

```
zSH> bridge add 1-6-1-701/gponport gtp 2 downlink-voice vlan 200 tagged
Adding bridge on 1-6-1-701/gponport
Created bridge-interface-record 1-6-1-701-gponport-200/bridge
```

Configuring an uplink bridge and downlink bridge on a GEM port for video services

Create an uplink and downlink bridge on a GEM port for video services.

See [Video Configuration on page 427](#) for complete details on creating bridged video.

- 1 Create the tagged uplink bridge with a VLAN ID.

```
zSH> bridge add 1-a-4-0/eth uplink vlan 300 tagged
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-300/bridge
Bridge-path added successfully
```

- 2 Modify the bridge path for the uplink bridge to set the multicast aging period and IGMP query interval.

```
zSH> bridge-path modify ethernet4-300/bridge vlan 300 default mcast 90 igmptimer 30
Bridge-path ethernet4-300/bridge/3/300/0/0/0/0/0/0/0 has been modified
```

- 3 Create the GPON traffic profile for the downlink bridge for video services.

```
zSH> new gpon-traffic-profile 3
gpon-traffic-profile 3
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}: 512
traffic-class: -----> {ubr}: cbr
compensated: -----> {false}: true
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: ----> {0}:
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

- 4 Create the downlink bridge with the GPON traffic profile and VLAN 300 and add the maximum video streams using the m/n format.

```
zSH> bridge add 1-6-1-901/gponport gtp 3 downlink-video vlan 300 tagged video 0/3
Adding bridge on 1-6-1-901/gponport
Created bridge-interface-record 1-6-1-901-gponport-300/bridge
```

Verify the configuration

Verify the configuration.

- 1 Verify the uplink and downlink bridges.

```
zSH> bridge show
```

Type	Orig	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-dat		Tagged	100	1/6/1/1/gpononu	1-6-1-501-gponport-100/bridge	DWN	
dwn-voi		Tagged	200	1/6/1/1/gpononu	1-6-1-701-gponport-200/bridge	DWN	

```

dwn-vid      Tagged 300 1/6/1/1/gpononu 1-6-1-901-gponport-300/bridge DWN
upl          Tagged 100 1/a/4/0/eth ethernet4-100/bridge DWN S VLAN 100 default
upl          Tagged 200 1/a/4/0/eth ethernet4-200/bridge DWN S VLAN 200 default
upl          Tagged 300 1/a/4/0/eth ethernet4-300/bridge DWN S VLAN 300 default
6 Bridge Interfaces displayed

```

2 Verify the GEM ports and their associated traffic profiles for the ONU.

zSH> **gpononu gempports 6/1/1**

								Fixed UBR	Fixed CBR	Assured
Max	Extra			traf				Bandwidth	Bandwidth	Bandwidth
Bandwidth	Bandwidth									
ONU	GEM Port	Admin	prof	compn	share	Mbits/sec	Mbits/sec	Mbits/sec		
Mbits/sec	Type	allocId	DBA							
=====	=====	=====	=====	=====	=====	=====	=====	=====	=====	=====
=====	=====	=====	=====	=====	=====	=====	=====	=====	=====	=====
a	n/a	1-6-1-1	1-6-1-501	Up	1	True	False	0	0.512	n/
		n/a	- n/a							
a	n/a	1-6-1-901	1-6-1-901	Up	3	True	False	0	0.512	n/
		n/a	- n/a							
a	n/a	1-6-1-701	1-6-1-701	Up	2	True	False	0	0.512	n/
		n/a	- n/a							

Advanced bridged data on the MXK with VLAN translation

This section discusses VLAN translation for bridged data on the MXK:

- [Overview of VLAN translation on the MXK, page 298](#)
- [Basic VLAN translation on bridges, page 299](#)
- [Advanced VLAN translation on bridges, page 303](#)

Overview of VLAN translation on the MXK

In situations when devices in the core network expect unique identifiers for each subscriber, and because subscriber configurations on the MXK can include large numbers of CPE devices with pre-configured VLAN IDs or VLAN/SLAN IDs, the MXK supports VLAN and SLAN translation from the subscriber to the MXK for VLAN/SLANs sent to the core network.

When configuring bridges for VLAN/SLAN translation, all network facing Ethernet ports *must* be *tagged* or *tagged* and all bridges facing the subscriber's CPE must be *tagged* or *tagged*. Bridges that are *untagged* do not support translation. For VLAN translation to work, there must be a VLAN or VLAN/SLAN in the Ethernet packet when it arrives at the MXK from the subscriber.

In cases where upstream devices in the core network from the MXK expect SLAN IDs, SLAN IDs can be promoted from downstream bridges to upstream bridges or translated if the subscriber traffic is already double-tagged.

For SLAN promotion *and* VLAN translation bridging configurations on the MXK, the name of the *tagged* bridge interface will include the interface, the translated to VLAN ID, and the SLAN ID. For example,

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 100 xlate-to 501 slan 1000 tagged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-501-1000/bridge
```

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
dwn-dat  100/----  Ty  501/1000  1/6/1/0/eth  1-6-1-0-eth-501-1000/bridge  UP
1 Bridge Interfaces displayed
```

The range for translated VLAN IDs is 1-4090 (some VLANs are reserved).

VLAN translation and VLAN translation and promotion is supported on Ethernet (single-slot only) and VDSL2.

Possible bridging configuration behaviors for VLAN/SLAN translation

Possible bridging configuration behaviors for VLAN/SLAN translation:

- either the network facing or the subscriber facing bridge is *untagged*
VLAN translation not allowed.
- subscriber facing single-tagged bridge to network facing single-tagged bridge with VLAN translation (*tagged* to *tagged*)
Refer to [VLAN translation on TLS bridges on page 299](#) and [VLAN translation on asymmetric bridges on page 301](#).
- subscriber facing single-tagged bridge to network facing double-tagged bridge with VLAN translation and SLAN promotion (*tagged* to *staged*)
Refer to [VLAN translation and SLAN promotion on asymmetric bridges on page 303](#).
- subscriber facing double-tagged bridge to network facing double-tagged bridge with SLAN translation (outer tag) (*staged* to *staged*)
Refer to [Configure asymmetric bridges with SLAN translation \(outer tag\) on page 306](#).
- subscriber facing double-tagged bridge to network facing double-tagged bridge with VLAN and SLAN translation (*staged* to *staged*)
Refer to [Configure asymmetric bridges for VLAN and SLAN translation on page 308](#).

bridge show command for VLAN translation

The **bridge show** command displays both subscriber facing VLAN/SLAN IDs and the translated network facing VLAN/SLAN IDs.

Basic VLAN translation on bridges

This section describes VLAN translation on both single-tagged TLS bridges and single-tagged asymmetrical bridges:

- [VLAN translation on TLS bridges, page 299](#)
- [VLAN translation on asymmetric bridges, page 301](#)

VLAN translation on TLS bridges

This section describes configuring TLS bridges on the MXK for basic VLAN translation.

When configuring the TLS bridges for VLAN translation, you *must* designate the TLS bridges as *tagged* on both the uplink Ethernet ports and the subscriber facing Ethernet ports. This allows the original VLAN ID on the subscriber side to pass down to the CPE, and the translated VLAN ID on the network side to pass to the core network.

As shown in [Figure 37](#), the VLAN ID 100 on the subscriber facing TLS bridges are translated on the MXK to VLAN ID 1001 for the network facing TLS bridge.

Figure 37: Single tagged to single tagged TLS bridges with VLAN ID translation



Configuring single-tagged to single-tagged TLS bridges with VLAN ID translation

- 1 Create a *tagged* TLS bridge on the network facing Ethernet port with VLAN ID.

```
zSH> bridge add 1-a-5-0/eth tls vlan 1001 tagged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-1001/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table
Data
-----
tls          Tagged 1001  1/a/5/0/eth  ethernet5-1001/bridge  UP
1 Bridge Interfaces displayed
```

- 2 Create *tagged* TLS bridges with the subscriber facing VLAN ID and the **xlate-to** VLAN ID on subscriber facing Ethernet ports.

```
zSH> bridge add 1-6-1-0/eth tls vlan 100 xlate-to 1001 tagged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-1001/bridge

zSH> bridge add 1-6-2-0/eth tls vlan 100 xlate-to 1001 tagged
Adding bridge on 1-6-2-0/eth
Created bridge-interface-record 1-6-2-0-eth-1001/bridge
```

Verify the TLS bridges. The **bridge show** command displays the VLAN ID of the downlink bridge(s) and the VLAN ID the MXK translated.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
tls  100      Tagged 1001  1/6/1/0/eth  1-6-1-0-eth-1001/bridge  UP  D 00:01:47:31:dc:1a
tls  100      Tagged 1001  1/6/2/0/eth  1-6-2-0-eth-1001/bridge  DWN
tls          Tagged 1001  1/a/5/0/eth  ethernet5-1001/bridge  UP
3 Bridge Interfaces displayed
```


Deleting single-tagged to single-tagged TLS bridges with VLAN translation

- 1 View the TLS bridges with VLAN ID translation.

```
zSH> bridge show
```

		Orig		Physical	Bridge	St	Table Data
Type	VLAN/SLAN	VLAN/SLAN					
tls	100	Tagged	1001	1/6/1/0/eth	1-6-1-0-eth-1001/bridge	UP	D 00:01:47:31:dc:1a
tls	100	Tagged	1001	1/6/2/0/eth	1-6-2-0-eth-1001/bridge	DWN	
tls		Tagged	1001	1/a/5/0/eth	ethernet5-1001/bridge	UP	

3 Bridge Interfaces displayed

- 2 Delete the TLS bridge on the Ethernet uplink port.

```
zSH> bridge delete ethernet5-1001/bridge vlan 1001
ethernet5-1001/bridge delete complete
```

- 3 Delete the TLS bridges on the Ethernet subscriber facing Ethernet ports. Bridges with VLAN ID translation use the translated VLAN ID in the **bridge delete** syntax.



Note: The VLAN ID added is *different* from the VLAN ID deleted.

```
zSH> bridge delete 1-6-2-0-eth-1001/bridge
Bridge-path deleted successfully
1-6-2-0-eth-1001/bridge delete complete
```

```
zSH> bridge delete 1-6-2-0-eth-1001/bridge
Bridge-path deleted successfully
1-6-2-0-eth-1001/bridge delete complete
```

VLAN translation on asymmetric bridges

This section describes configuring asymmetric bridges on the MXK for basic VLAN translation.

When configuring the MXK for VLAN translation on asymmetric bridges, you *must* designate the uplink bridge as *tagged* to pass the translated VLAN ID to the core network and the downlink bridge as *tagged* to pass the original VLAN ID down to the subscriber.

As shown in [Figure 38](#), the VLAN ID 100 on subscriber facing downlink bridges are translated on the MXK to VLAN ID 1002 for the network facing uplink bridge.

Figure 38: Asymmetric bridges with VLAN translation



Configuring single-tagged to single-tagged asymmetric bridges for VLAN translation

- 1 Create a *tagged* uplink bridge with VLAN ID on the network facing Ethernet port.

```
zSH> bridge add 1-a-4-0/eth uplink vlan 1002 tagged
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-1002/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
upl          Tagged 1002  1/a/4/0/eth  ethernet4-1002/bridge  DWN S VLAN 1002 default
1 Bridge Interfaces displayed
```

- 2 Create *tagged* downlink bridges with the subscriber facing VLAN ID and the **xlate-to** VLAN ID on subscriber facing Ethernet ports.

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 100 xlate-to 1002 tagged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-1002/bridge

zSH> bridge add 1-6-2-0/eth downlink-data vlan 100 xlate-to 1002 tagged
Adding bridge on 1-6-2-0/eth
Created bridge-interface-record 1-6-2-0-eth-1002/bridge
```

Verify the downlink bridges. The **bridge show** command displays the VLAN ID of the downlink bridge(s) and the VLAN ID the MXK translated.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
dwn-dat  100      Tagged 1002  1/6/1/0/eth  1-6-1-0-eth-1002/bridge  UP  D 00:01:47:31:dc:1a
dwn-dat  100      Tagged 1002  1/6/2/0/eth  1-6-2-0-eth-1002/bridge  DWN
upl          Tagged 1002  1/a/4/0/eth  ethernet4-1002/bridge  DWN S VLAN 1002 default
3 Bridge Interfaces displayed
```

Deleting single-tagged to single-tagged asymmetric bridges with VLAN ID translation

- 1 View the existing bridges.

```
zSH> bridge show
```

Type	Orig	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
dwn-dat	100	Tagged	1002	1/6/1/0/eth	1-6-1-0-eth-1002/bridge	UP	D	00:01:47:31:dc:1a
dwn-dat	100	Tagged	1002	1/6/2/0/eth	1-6-2-0-eth-1002/bridge	DWN		
upl		Tagged	1002	1/a/4/0/eth	ethernet4-1002/bridge	DWN	S	VLAN 1002 default

3 Bridge Interfaces displayed

- 2 Delete the uplink bridge.

```
zSH> bridge delete ethernet4-1002/bridge vlan 1002
```

```
Bridge-path deleted successfully
```

```
ethernet4-1002/bridge delete complete
```

- 3 Delete the downlink bridge. Bridges with VLAN ID translation use the translated VLAN ID in the **bridge delete** syntax.



Note: The VLAN ID added is *different* from the VLAN ID deleted.

```
zSH> bridge delete 1-6-1-0-eth-1002/bridge vlan 1002
```

```
1-6-1-0-eth-1002/bridge delete complete
```

```
zSH> bridge delete 1-6-2-0-eth-1002/bridge vlan 1002
```

```
1-6-2-0-eth-1002/bridge delete complete
```

Advanced VLAN translation on bridges

This section includes the following topics:

- [VLAN translation and SLAN promotion on asymmetric bridges, page 303](#)
- [Configure asymmetric bridges with SLAN translation \(outer tag\), page 306](#)
- [Configure asymmetric bridges for VLAN and SLAN translation, page 308](#)
- [VLAN translation on Active Ethernet asymmetric bridges with CoS replacement, page 311](#)

VLAN translation and SLAN promotion on asymmetric bridges

This section describes configuring asymmetric bridges on the MXK for VLAN translation and SLAN promotion.

When configuring uplink and downlink bridges for VLAN translation and SLAN promotion, the uplink bridges are *tagged* and the downlink bridges are *tagged*. This will pass the translated VLAN ID and promoted SLAN ID to the network. On the downlink bridge the original VLAN passes down to the subscriber.

For this type of configuration on the MXK, when a bridge is configured for both VLAN translation and SLAN promotion, the name of the *tagged* bridge interface will include the SLAN ID.



Note: This feature is valid on single-slot Ethernet cards and VDSL2 combo cards with splitter.

For example,

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 100 xlate-to 501 slan 1000 tagged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-501-1000/bridge

zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge                                     St  Table Data
-----
dwn-dat   100/----  Tg  501/1000  1/6/1-0/eth  1-6-1-0-eth-501-1000/bridge             UP  D 00:01:47:31:dc:1a
1 Bridge Interfaces displayed
```

As shown in [Figure 39](#), the VLAN ID100 on subscriber facing downlink bridges are translated on the MXK to unique VLAN IDs for the uplink bridge and SLAN ID 500 is promoted to the uplink.

In this configuration, the uplink bridge is configured with VLAN ID 0, a wildcard, to accept all VLAN IDs to send to the core network.

Figure 39: Asymmetric bridges with VLAN translation and SLAN promotion



Configuring single-tagged to double-tagged asymmetric bridges with VLAN translation and SLAN promotion

- 1 Create the *tagged* uplink bridge with VLAN ID 0 (accepts all VLANs) and SLAN ID 500.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 0 slan 500 tagged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-0-500/bridge
```

Bridge-path added successfully

Verify the bridge.

```
zSH> bridge show
```

Type	VLAN/SLAN	Orig VLAN/SLAN	Physical	Bridge	St	Table Data
upl		ST 0/500	1/a/5/0/eth	ethernet5-0-500/bridge	UP	S SLAN 500 VLAN 0 default

1 Bridge Interfaces displayed

- 2 Create *tagged* downlinks with VLAN ID, the **xlate-to** VLAN ID, and the SLAN ID for network promotion.

Designating *tagged* does not pass the SLAN ID to the CPE.

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 100 xlate-to 1001 slan 500 tagged
```

Adding bridge on 1-6-1-0/eth

Created bridge-interface-record 1-6-1-0-eth-1001-500/bridge

```
zSH> bridge add 1-6-2-0/eth downlink-data vlan 100 xlate-to 1002 slan 500 tagged
```

Adding bridge on 1-6-2-0/eth

Created bridge-interface-record 1-6-2-0-eth-1002-500/bridge

```
zSH> bridge add 1-6-3-0/eth downlink-data vlan 100 xlate-to 1003 slan 500 tagged
```

Adding bridge on 1-6-3-0/eth

Created bridge-interface-record 1-6-3-0-eth-1003-500/bridge

Verify the bridge. The **bridge show** command displays the VLAN ID of the downlink bridge(s) and the VLAN ID the MXK translated.

```
zSH> bridge show
```

Type	VLAN/SLAN	Orig VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-dat	100/----	Tg 1001/500	1/6/1/0/eth	1-6-1-0-eth-1001-500/bridge	UP	D 00:01:47:31:dc:1a
dwn-dat	100/----	Tg 1002/500	1/6/2/0/eth	1-6-2-0-eth-1002-500/bridge	DWN	
dwn-dat	100/----	Tg 1003/500	1/6/3/0/eth	1-6-3-0-eth-1003-500/bridge	DWN	
upl		ST 0/500	1/a/5/0/eth	ethernet5-0-500/bridge	UP	S SLAN 500 VLAN 0 default

4 Bridge Interfaces displayed

Deleting single-tagged to double-tagged asymmetric bridges with VLAN translation and SLAN promotion

- 1 View the existing bridges.

```
zSH> bridge show
```

Type	VLAN/SLAN	Orig VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-dat	100/----	Tg 1001/500	1/6/1/0/eth	1-6-1-0-eth-1001-500/bridge	UP	D 00:01:47:31:dc:1a
dwn-dat	100/----	Tg 1002/500	1/6/2/0/eth	1-6-2-0-eth-1002-500/bridge	DWN	
dwn-dat	100/----	Tg 1003/500	1/6/3/0/eth	1-6-3-0-eth-1003-500/bridge	DWN	
upl		ST 0/500	1/a/5/0/eth	ethernet5-0-500/bridge	UP	S SLAN 500 VLAN 0 default

4 Bridge Interfaces displayed

2 Delete the uplink bridge.

```
zSH> bridge delete ethernet5-0-500/bridge
Bridge-path deleted successfully
ethernet5-0-500/bridge delete complete
```

3 Delete the downlink bridges. Bridges with VLAN ID translation use the translated VLAN ID in the **bridge delete** syntax.



Note: The VLAN ID added is *different* from the VLAN ID deleted.

```
zSH> bridge delete 1-6-1-0-eth-1001-500/bridge vlan 1001
1-6-1-0-eth-1001-500/bridge delete complete
```

```
zSH> bridge delete 1-6-2-0-eth-1002-500/bridge vlan 1002
1-6-2-0-eth-1002-500/bridge delete complete
```

```
zSH> bridge delete 1-6-3-0-eth-1003-500/bridge vlan 1003
1-6-3-0-eth-1003-500/bridge delete complete
```

Configure asymmetric bridges with SLAN translation (outer tag)

This section describes configuring asymmetric bridges on the MXK for SLAN translation (outer tag).

In certain cases it may be necessary to translate double-tagged CPE downstream devices configured with the same SLAN IDs to uplink bridges configured with different SLAN IDs. The uplink bridges are *staged* and the downlink bridges are also *staged* because the CPE device is expecting an SLAN ID.

As shown in [Figure 40](#), the VLAN ID 200 is passed from the downlink to the uplink, and the SLAN ID 1000 is translated on the MXK for the network facing uplink bridge.

Figure 40: Asymmetric bridges with SLAN (outer tag) translation



Configuring double-tagged to double-tagged asymmetric bridges for SLAN translation

1 Create *staged* uplink bridges with VLAN ID and SLAN ID which are sent to the network.

```
zSH> bridge add 1-a-4-0/eth uplink vlan 200 slan 1001 staged
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-200-1001/bridge
Bridge-path added successfully
```

```
zSH> bridge add 1-a-5-0/eth uplink vlan 200 slan 1002 staged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-200-1002/bridge
Bridge-path added successfully
```

Verify the bridges.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
upl          ST 200/1001  1/a/4/0/eth  ethernet4-200-1001/bridge  DWN S SLAN 1001 VLAN 200
default
upl          ST 200/1002  1/a/5/0/eth  ethernet5-200-1002/bridge  UP  S SLAN 1002 VLAN 200
default
2 Bridge Interfaces displayed
```

2 Create the *staged* downlink bridges with VLAN ID and the **xlate-to** SLAN ID.

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 200 slan 1000 xlate-to 1001
staged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-200-1001/bridge
```

```
zSH> bridge add 1-6-2-0/eth downlink-data vlan 200 slan 1000 xlate-to 1002 staged
Adding bridge on 1-6-2-0/eth
Created bridge-interface-record 1-6-2-0-eth-200-1002/bridge
```

Verify the bridge. The **bridge show** command displays the VLAN ID of the downlink bridge(s) and the SLAN ID the MXK translated.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
dwn-dat ---/1000 ST 200/1001  1/6/1/0/eth  1-6-1-0-eth-200-1001/bridge  UP
dwn-dat ---/1000 ST 200/1002  1/6/2/0/eth  1-6-2-0-eth-200-1002/bridge  DWN
upl          ST 200/1001  1/a/4/0/eth  ethernet4-200-1001/bridge  DWN S SLAN 1001 VLAN 200 default
upl          ST 200/1002  1/a/5/0/eth  ethernet5-200-1002/bridge  UP  S SLAN 1002 VLAN 200 default
4 Bridge Interfaces displayed
```

Deleting double-tagged to double-tagged on asymmetric bridges with SLAN translation

1 View the bridges.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
```

```

dwn-dat ----/1000 ST 200/1001 1/6/1/0/eth 1-6-1-0-eth-200-1001/bridge UP
dwn-dat ----/1000 ST 200/1002 1/6/2/0/eth 1-6-2-0-eth-200-1002/bridge DWN
upl          ST 200/1001 1/a/4/0/eth ethernet4-200-1001/bridge DWN S SLAN 1001 VLAN 200 default
upl          ST 200/1002 1/a/5/0/eth ethernet5-200-1002/bridge UP S SLAN 1002 VLAN 200 default
4 Bridge Interfaces displayed

```

2 Delete the uplink bridges.

```

zSH> bridge delete ethernet4-200-1002/bridge vlan 200
Bridge-path deleted successfully
ethernet4-200-1002/bridge delete complete

```

```

zSH> bridge delete ethernet5-200-1001/bridge vlan 200
Bridge-path deleted successfully
ethernet5-200-1001/bridge delete complete

```

3 Delete the downlink bridges.

```

zSH> bridge delete 1-6-1-0-eth-200-1001/bridge vlan 200
1-6-1-0-eth-200-1001/bridge delete complete

```

```

zSH> bridge delete 1-6-2-0-eth-200-1002/bridge vlan 200
1-6-2-0-eth-200-1002/bridge delete complete

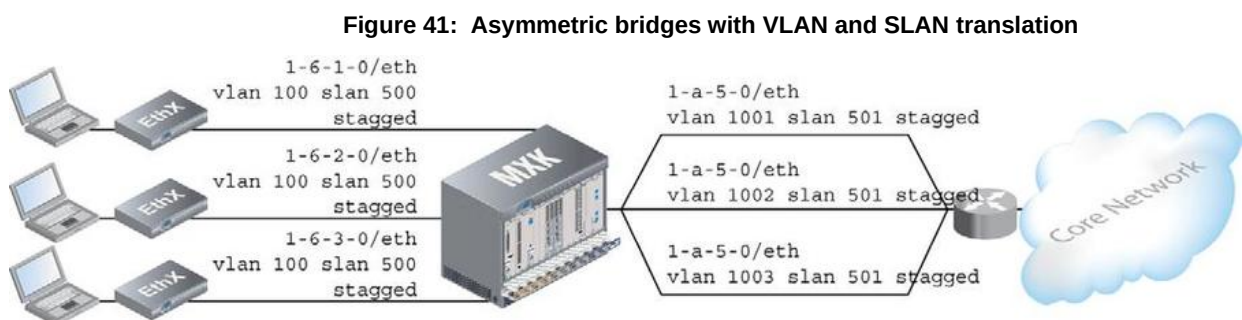
```

Configure asymmetric bridges for VLAN and SLAN translation

This section describes configuring asymmetric bridges on the MXK for VLAN and SLAN ID translation. This configuration can be used in situations where CPE devices are configured with the same VLAN ID and SLAN ID and need to connect with existing networks.

When configuring the uplink and the downlink bridges for VLAN and SLAN translation, both bridges are *staged* to allow the VLAN ID and the SLAN ID to pass to the downstream CPE and the MXK translated VLAN ID and SLAN ID to pass to the core network.

As shown in [Figure 41](#), the VLAN ID 100 and the SLAN 500 ID are translated by the MXK for various uplink bridges.



Configuring double-tagged to double-tagged bridges for VLAN and SLAN translation

- 1 Create *staged uplink* bridges for the MXK translated VLAN ID and SLAN ID to send to the core network.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 1001 slan 501 staged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-1001-501/bridge
Bridge-path added successfully
```

```
zSH> bridge add 1-a-5-0/eth uplink vlan 1002 slan 502 staged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-1002-502/bridge
Bridge-path added successfully
```

```
zSH> bridge add 1-a-5-0/eth uplink vlan 1003 slan 503 staged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-1003-503/bridge
Bridge-path added successfully
```

Verify the bridges.

```
zSH> bridge show
```

Type	VLAN/SLAN	Physical	Bridge	St	Table	Data
upl	ST 1001/501	1/a/5/0/eth	ethernet5-1001-501/bridge	UP	S	SLAN 501 VLAN 1001 default
upl	ST 1002/502	1/a/5/0/eth	ethernet5-1002-502/bridge	UP	S	SLAN 502 VLAN 1002 default
upl	ST 1003/503	1/a/5/0/eth	ethernet5-1003-503/bridge	UP	S	SLAN 503 VLAN 1003 default

3 Bridge Interfaces displayed

- 2 Create *staged downlink* bridges with the VLAN ID and SLAN ID and the **xlate-to** VLAN ID and the SLAN ID.

```
zSH> bridge add 1-6-1-0/eth downlink-data vlan 100 xlate-to 1001 slan 500
xlate-to 501 staged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-1001-501/bridge
```

```
zSH> bridge add 1-6-2-0/eth downlink-data vlan 100 xlate-to 1002 slan 500
xlate-to 502 staged
Adding bridge on 1-6-2-0/eth
Created bridge-interface-record 1-6-2-0-eth-1002-501/bridge
```

```
zSH> bridge add 1-6-3-0/eth downlink-data vlan 100 xlate-to 1003 slan 500
xlate-to 503 staged
Adding bridge on 1-6-3-0/eth
Created bridge-interface-record 1-6-3-0-eth-1003-502/bridge
```

Verify the bridges. The **bridge show** command displays the VLAN/SLAN IDs of the downlink bridge(s) and the VLAN/SLAN IDs the MXK translated.

```
zSH> bridge show
```

Type	Orig		Physical	Bridge	St	Table Data
	VLAN/SLAN	VLAN/SLAN				
dwn-dat	100/500	ST 1001/501	1/6/1/0/eth	1-6-1-0-eth-1001-501/bridge	UP	
dwn-dat	100/500	ST 1002/502	1/6/2/0/eth	1-6-2-0-eth-1002-502/bridge	DWN	
dwn-dat	100/500	ST 1003/503	1/6/3/0/eth	1-6-3-0-eth-1003-503/bridge	DWN	
upl		ST 1001/501	1/a/5/0/eth	ethernet5-1001-501/bridge	UP	S SLAN 501 VLAN 1001 default
upl		ST 1002/502	1/a/5/0/eth	ethernet5-1002-502/bridge	UP	S SLAN 502 VLAN 1002 default
upl		ST 1003/503	1/a/5/0/eth	ethernet5-1003-503/bridge	UP	S SLAN 503 VLAN 1003 default

6 Bridge Interfaces displayed

Deleting double-tagged to double-tagged bridges with VLAN and SLAN translation

1 Verify the bridges.

zSH> **bridge show**

Type	Orig		Physical	Bridge	St	Table Data
	VLAN/SLAN	VLAN/SLAN				
dwn-dat	100/500	ST 1001/501	1/6/1/0/eth	1-6-1-0-eth-1001-501/bridge	UP	
dwn-dat	100/500	ST 1002/502	1/6/2/0/eth	1-6-2-0-eth-1002-502/bridge	DWN	
dwn-dat	100/500	ST 1003/503	1/6/3/0/eth	1-6-3-0-eth-1003-503/bridge	DWN	
upl		ST 1001/501	1/a/5/0/eth	ethernet5-1001-501/bridge	UP	S SLAN 501 VLAN 1001 default
upl		ST 1002/502	1/a/5/0/eth	ethernet5-1002-502/bridge	UP	S SLAN 502 VLAN 1002 default
upl		ST 1003/503	1/a/5/0/eth	ethernet5-1003-503/bridge	UP	S SLAN 503 VLAN 1003 default

6 Bridge Interfaces displayed

2 Delete the uplink bridges.

zSH> **bridge delete ethernet5-1001-501/bridge vlan 1001**
 Bridge-path deleted successfully
 ethernet5-1001-501/bridge delete complete

zSH> **bridge delete ethernet5-1002-502/bridge vlan 1002**
 Bridge-path deleted successfully
 ethernet5-1002-502/bridge delete complete

zSH> **bridge delete ethernet5-1003-503/bridge vlan 1003**
 Bridge-path deleted successfully
 ethernet5-1003-503/bridge delete complete

3 Delete the downlink bridges. Bridges with VLAN ID translation use the translated VLAN ID in the **bridge delete** syntax.



Note: The VLAN ID added is *different* from the VLAN ID deleted.

zSH> **bridge delete 1-6-1-0-eth-1001-501/bridge vlan 1001**
 1-6-1-0-eth-1001-501/bridge delete complete

zSH> **bridge delete 1-6-2-0-eth-1002-502/bridge vlan 1002**
 1-6-2-0-eth-1002-502/bridge delete complete

zSH> **bridge delete 1-6-3-0-eth-1003-503/bridge vlan 1003**
 1-6-3-0-eth-1003-503/bridge delete complete

VLAN translation on Active Ethernet asymmetric bridges with CoS replacement

When VLAN translation is provided on Active Ethernet downlink bridges, CoS replacement may be provided as well. On traffic which is coming from the downstream subscriber side, the CoS bit may be changed to a different CoS in the upstream traffic.

The **cos** keyword with a value of **1** to **7** in the **bridge add** command sets the CoS value regardless of the CoS value which was set downstream. If the **cos** keyword is set to **0**, the CoS value will pass through without being changed.

Figure 42: Asymmetric bridges with VLAN translation and CoS replacement



Configure single-tagged to single-tagged asymmetric bridges for VLAN translation with CoS

When configuring the MXK for VLAN translation on asymmetric bridges, you *must* designate the uplink bridge as *tagged* to pass the translated VLAN ID to the core network and the downlink bridge as *tagged* to pass the original VLAN ID down to the subscriber.

To add the CoS replacement use the **bridge add** command to configure a CoS value on an Active Ethernet downlink bridge configured for VLAN translation. Use the **cos** keyword to configure the CoS replacement value on the downlink per bridge interface.

As shown in [Figure 14](#), the VLAN ID 100 on subscriber facing downlink bridges is translated on the MXK to VLAN ID 1002 for the network facing uplink bridge. The CoS value of 5 is inserted into the priority bit of the Ethernet frame on ingress.

Configuring single-tagged to single-tagged asymmetric bridges for VLAN translation with CoS

- 1 Create a *tagged* uplink bridge with VLAN ID on the network facing Ethernet port.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 1002 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-1002/bridge
Bridge-path added successfully
```

- 2 Create a *tagged* downlink bridge with the subscriber facing VLAN ID, the **xlate-to** VLAN ID, and the CoS replacement value.

```
zSH> bridge add 1-6-5-0/eth downlink-data vlan 100 xlate-to 1002 tagged cos 5
Adding bridge on 1-6-5-0/eth
Created bridge-interface-record 1-6-5-0-eth-1002/bridge
```

Verify the bridge interfaces. The **bridge show** command displays the VLAN ID of the downlink bridge and the VLAN ID the MXK translated.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge
-----
dwn-dat   100        Tagged 1002  1/6/5/0/eth  1-6-5-0-eth-1002/bridge  DWN
upl                Tagged 1002  1/a/2/0/eth  ethernet2-1002/bridge    DWN S VLAN 1002 default
2 Bridge Interfaces displayed
```



Note: The **cos** value of **0** in the **bridge add** command with **xlate-to** means that the CoS value from the downstream traffic will not be altered.

Deleting single-tagged to single-tagged asymmetric bridges with VLAN ID translation

- 1 View the existing bridges.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge
-----
upl                Tagged 1002  1/a/2/0/eth  ethernet2-1002/bridge    DWN S VLAN 1002 default
dwn   100        Tagged 1002  1/6/5/0/eth  1-6-5-0-eth-1002/bridge  DWN
1 Bridge Interfaces displayed
```

- 2 Delete the uplink bridge.

```
zSH> bridge delete ethernet2-1002/bridge
```

- 3 Delete the downlink bridge. Bridges with VLAN ID translation use the translated VLAN ID in the **bridge delete** syntax.

```
zSH> bridge delete 1-6-5-0-eth-1002/bridge
```

Filters for MXK bridges (packet-rule-record)

This section explains how to configure **packet-rule-record** filters and includes:

- [Overview of packet-rule-record filters, page 313](#)
- [Option 82 DHCP on bridge packet rule \(bridgeinsertoption82\), page 316](#)
- [DHCP on bridge packet rules \(DHCP relay, and Forbid OUI\), page 323](#)
- [PPPoE with intermediate agent \(bridgeinsertpppoevendortag\), page 327](#)
- [Bandwidth limiting by port and service, single and dual rate limiting, page 334](#)
- [Destination MAC swapping, page 352](#)
- [Bridge storm protection, page 355](#)
- [Access Control List \(ACL\), page 367](#)

Overview of packet-rule-record filters

The SLMS CLI architecture has a mechanism for adding one or more filters to the ingress and egress bridge interfaces by grouping **packet-rule-record(s)**.

Multiple bridges may use the same packet rule group/index as shown in [Figure 43](#).

Figure 43: Multiple filters for bridge interfaces



Create packet-rule-record filters

Use the **rule add** command to create a packet rule by entering the group index and the member index when you create the rule.

The **bridge-interface-record** accesses rules by the group index number.

```
rule add <ruleType> <groupIndex/memberIndex> <value [value] [value]...>
```

The **packetRuleValue** options depend on the **packetRuleType** selected.

For example, the rule *bridgeforbidoui* **packetRuleType**, requires the first three bytes of the MAC address in the value field. For example,

```
zSH> rule add bridgeforbidoui 4/1 AA:BB:CC
Created packet-rule-record 4/1 (bridgeforbidoui)
```

In the case of the rule *ratelimitdiscard* **packetRuleType**, dual rate limiting requires two options, a committed rate and a peak rate. For example,

```
zSH> rule add ratelimitdiscard 5/1 rate 2000 peak 4000
Created packet-rule-record 5/1 (ratelimitdiscard)
```

The **bridge add** command uses the variables **ipktrule** or **epktrule** to reference the group number. Entering **ipktrule** adds the filter on the bridge ingress and **epktrule** adds the filter on the bridge egress.

Bridge interfaces can be configured with **ipktrule** or **epktrule**, or both.

For example:

```
zSH> bridge add 1-6-1-0/eth downlink vlan 777 ipktrule 4 epktrule 5
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth/bridge
```

Creating a packet rule group index with packet rule records

- 1 Use the **rule add** command to add a rule type to a group and member index and the parameter(s) which define that rule type.

This example creates a **packet-rule-group** index with two members.

The **dstmacswappingstatic** rule shown requires a parameter which is a MAC address. Entering *ipktrule* will enter the rules on the ingress of the bridge.

```
zSH> rule add dstmacswapstatic 2/1 08:00:20:bc:8b:8c
Created packet-rule-record 2/1 (dstmacswapstatic)
```

Add another rule to the group index, if needed.

```
zSH> rule add bridgedhcprelay 2/2 20
Created packet-rule-record 2/2 (bridgedhcprelay)
```

Display the **packet-rule-group** with members.

```

zSH> rule show

```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30
		auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30
		auto-enable-interval (def) 300 600 1200
2/1	dstmacswapstatic	08:00:20:bc:8b:8c
2/2	bridgedhcprelay	20

```

4 record(s) found

```

2 Create the bridge and include the IP packet rule group

```

zSH> bridge add 1-6-1-0/eth downlink vlan 777 ipktrule 2
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth/bridge

```

Deleting a packet rule

Use the **rule delete** command to delete the rule from the group index.

```

zSH> rule delete 2/1
packet-rule-record 2/1 Deleted completely

zSH> rule delete 2/2
packet-rule-record 2/2 Deleted completely

```

Packet rule types

Packet rules types on the MXK:

- *bridgeinsertoption82*
Insert DHCP option 82 parameter.
See [DHCP on bridge packet rules \(DHCP relay, and Forbid OUI\) on page 323](#)
- *bridgedhcprelay*
Enables DHCP relay.
See [DHCP on bridge packet rules \(DHCP relay, and Forbid OUI\) on page 323](#)
- *bridgeforbidoui*
Forbid OUI.
See [DHCP on bridge packet rules \(DHCP relay, and Forbid OUI\) on page 323](#)
- *bridgeinsertpppoevendortag*
See [PPPoE with intermediate agent \(bridgeinsertpppoevendortag\) on page 327](#)
- *dstmacswapdynamic*

destmacswapstatic

See [Destination MAC swapping on page 352](#).

- *ratelimitdiscard*

Discard packets in excess of rate (kbps)

colorawareratelimitdiscard

Discard packets in excess of rate (kbps) (color aware)

See [Bandwidth limiting by port and service, single and dual rate limiting on page 334](#).

- *promotefirstencapsulationvlan*

Defines the outer VLAN ID (third tag) for the access facing TLS bridge that will be promoted to the network for Q-in-Q-in-Q.

filterfirstencapsulationvlan

Defines the outer VLAN ID tag that will be stripped going to the access TLS bridge and inserted (promoted) to the network TLS bridge for Q-in-Q-in-Q.

See [Q-in-Q-in-Q \(VLAN IDs, SLAN IDs and packet rules\) on bridges on page 259](#).

- *bridgestormdetect*

Provides a way to analyze packets by capturing discarded packets when a certain threshold is reached and is configured only on the ingress of a bridge interface.

See [Bridge storm protection on page 355](#).

- *dscptocos*

See [DSCP to COS \(802.1p\) mapping on page 348](#).

- *allow, deny*

See [Access Control List \(ACL\) on page 367](#).

The ACL filters allow you to deny or allow packets based on packet characteristics.

Option 82 DHCP on bridge packet rule (bridgeinsertoption82)

This section covers the two methods used to configure the *bridgeinsertoption82* rule type and includes:

- [Option 82 for DHCP relay overview, page 317](#)
- [Option 82 DHCP on bridge packet rule \(bridgeinsertoption82\) configuration without macros defined strings, page 318](#)
- [Option 82 DHCP on bridge packet rule \(bridgeinsertoption82\) configuration with macro defined strings, page 319](#)

Option 82 for DHCP relay overview

When acting as a DHCP relay agent, the MXK includes option 82 to identify the requesting client to the DHCP server. There are two sub-options for DHCP option 82 insert, Circuit ID and Remote ID. Both of these fields are text fields, though they were designed to carry specific information.

You can define textual values for two items of textual information: circuit ID and remote ID.

If the first value is set it is taken as a literal text string to be used as the suboption 1 field in the DHCP packet. If it is not set a text string identifying the box and interface which received the packet is used. If the second value is set it is taken as a literal text string to be used as the suboption 2 field in the DHCP packet. If it is not set no suboption2 is provided.

Use of this feature will usually require a distinct rule group for each interface since the circuit and remote Id values associated with suboptions 1 and 2 are distinct for each interface.

Circuit ID is meant to provide information about the circuit which the request came in on. It is normally the port and interface information.

RFC 3046 describes possible uses of the Circuit ID field:

- Router interface number
- Remote Access Server port number
- Frame Relay DLCI
- ATM virtual circuit number
- Cable Data virtual circuit number

Remote ID is meant to provide information about the remote host end of the circuit, however in practice the sub-option usually contains information about the relay agent.

RFC 3046 describes possible uses of the Remote ID field:

- a "caller ID" telephone number for dial-up connection
- a "user name" prompted for by a Remote Access Server
- a remote caller ATM address
- a "modem ID" of a cable data modem
- the remote IP address of a point-to-point link
- a remote X.25 address for X.25 connections

Option 82 DHCP on bridge packet rule (bridgeinsertoption82) configuration without macros defined strings

The default information inserted into the packet during the DHCP discovery process is formatted as:

System 0_ip:IfName

The systemIP address is taken from the IP address configured in the **system 0** profile. If the IP address is not defined in the system 0 profile, *0.0.0.0* is inserted.

Creating a packet rule for bridgeinsertoption82 without macro defined strings

1 Create the *bridgeinsertoption82* filter for default information.

```
zSH> rule add bridgeinsertoption82 1/1
Created packet-rule-record 1/1 (bridgeinsertoption82)
```

2 Verify the rule.

```
zSH> rule show
```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30
		auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30
		auto-enable-interval (def) 300 600 1200
1/1	bridgeinsertoption82	

3 record(s) found

3 To specify the first **packetRuleType**:

```
zSH> rule add bridgeinsertoption82 2/1 oakland
Created packet-rule-record 2/1 (bridgeinsertoption82)
```

4 Verify the rule.

```
zSH> rule show
```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30
		auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30
		auto-enable-interval (def) 300 600 1200
1/1	bridgeinsertoption82	
2/1	bridgeinsertoption82	oakland

4 record(s) found

5 To specify only the second **packetRuleType**:

```
zSH> rule add bridgeinsertoption82 3/1 "" 510-555-1111
Created packet-rule-record 3/1 (bridgeinsertoption82)
```

6 Verify the rule.

```
zSH> rule show
```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30
		auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30
		auto-enable-interval (def) 300 600 1200
1/1	bridgeinsertoption82	
2/1	bridgeinsertoption82	oakland
3/1	bridgeinsertoption82	510-555-1111

5 record(s) found

7 To specify both values:

```
zSH> rule add bridgeinsertoption82 4/1 oakland 510-555-1111
Created packet-rule-record 4/1 (bridgeinsertoption82)
```

8 Verify the rule.

```
zSH> rule show
```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30
		auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30
		auto-enable-interval (def) 300 600 1200
1/1	bridgeinsertoption82	
2/1	bridgeinsertoption82	oakland
3/1	bridgeinsertoption82	510-555-1111
4/1	bridgeinsertoption82	oakland 510-555-1111

6 record(s) found

Option 82 DHCP on bridge packet rule (bridgeinsertoption82) configuration with macro defined strings

This section discusses how to insert customized strings with the use of supported macro formats as shown in [Table 22](#).

If the **packetRuleValue** field contains one or more dollar sign (\$) characters, the vendor tag text that would have been supplied is replaced by the contents of the field as follows:

- When a dollar sign character is encountered, the text following the dollar sign is compared to [Table 22](#).
- If no match is found the dollar sign character is replaced with the text "Unknown".
- If a match is found the dollar sign character and the associated text is replaced by the text indicated.
- The macro name and abbreviations are both case sensitive.

- The \$macro strings may be imbedded in literal text. This text is copied to the output without change.
- The supported macro formats may be entered in the text as either \$macroname or \$abbreviation. Thus \$SystemName and \$NM give the same result, which is to substitute the system name from the **system 0** profile.

Some of the macros vary in effect depending on the value they are intended to display.

- \$Gem and \$Onu IDs are displayed or not depending on whether or not they have a non-zero value.
- \$Vlan displays -SLAN-VLAN if the SLAN is non-zero, -VLAN if the -SLAN is zero but the VLAN is non-zero, or nothing if they are both zero.
- \$VC displays -vpi-vci if either value is non-zero and nothing if they are both zero.



Note: Macro names are case sensitive.

Table 21: Supported macro formats for macro defined strings

Macro name	Abbreviation	Varies	Result
\$SystemName	NM	NM	sysname from the system 0 profile.
\$SystemIP	IP	No	ipaddress address from the system 0 profile.
\$IfName	IF	IF	ifName from the bridge IfTranslate profile.
\$Address	AD	No	<i>shelf-slot-port-subport-type</i> of the underlying physical interface. Where the interface is a GPON OLT interface the type is given as <i>gponport</i> and the subport is the GEM port.
\$Shelf	SH	No	Shelf (currently always 1).
\$Slot	SL	No	<i>slot</i> from the IfTranslate profile of the underlying physical interface.
\$Port	PT	No	<i>port</i> (see \$Slot).
\$SubPort	SP	No	<i>subport</i> (see \$Slot.) For GPON this is the GEM port
\$Gem	GM	Yes	-GEMPort (or nothing)
\$Onu	ON	Yes	-ONUnumber (or nothing)
\$Type	TY	No	Type (for GPON this is <i>gponport</i>).
\$Vlan	VN	Yes	-SLAN-VLAN (or -VLAN or nothing).
\$Svlan	SV	No	SLAN

Table 21: Supported macro formats for macro defined strings (Continued)

Macro name	Abbreviation	Varies	Result
\$Cvlan	CV	No	VLAN
\$Vc	VC	Yes	-VPI-VCI (or nothing)
\$Vpi	VP	No	-VPI
\$Vci	VI	No	-VCI
\$Null	NL	No	Nothing (used to change PPPoE handling of constant strings).

Creating a packet rule for bridgeinsertoption82 with macro defined strings

Create a **packet-rule-record** using macro names to create a user-defined string. Strings created with macros, including the information pulled in by the macro, are limited to 48 characters.

- 1 To create a string for the first **packetRuleType** field:
 - a To create a string that includes system IP address, IfName (typically *shelf/slot/port/subport*), and VLAN ID for the first **packetRuleType** field, enter:

```
zSH> rule add bridgeinsertoption82 1/1 $SystemIP$IfName$Vlan
Created packet-rule-record 1/1 (bridgeinsertoption82)
```

The *\$SystemIP* macro looks in the **system 0** profile for the IP address and to the bridge configuration for the rest of the information.

View the **system 0** profile.

```
zSH> get system 0
system 0
syscontact: -----> {}
sysname: -----> {MXK -California}
syslocation: -----> {}
enableauthtraps: -----> {disabled}
setserialno: -----> {0}
zmsexists: -----> {true}
zmsconnectionstatus: --> {inactive}
zmsipaddress: -----> {172.24.84.105}
configsyncexists: -----> {false}
configsyncoverflow: ---> {false}
configsyncpriority: ---> {high}
configsyncaction: -----> {noaction}
configsyncfilename: ---> {}
configsyncstatus: -----> {synccomplete}
configsyncuser: -----> {cfgsync}
configsyncpasswd: -----> ** private **
numshelves: -----> {1}
shelvesarray: -----> {}
```

```

numcards: -----> {3}
ipaddress: -----> {172.16.160.49}
alternateipaddress: ---> {0.0.0.0}
countryregion: -----> {us}
primaryclocksource: ---> {0/0/0/0/0}
ringsource: -----> {internalringsourcelabel}
revertiveclocksource: -> {true}
voicebandwidthcheck: --> {false}
alarm-levels-enabled: -> {critical+major+minor+warning}
userauthmode: -----> {local}
radiusauthindex: -----> {0}
secure: -----> {disabled}
webinterface: -----> {enabled}
options: -----> {NONE(0)}
.....

```

b Verify the packet-rule-record.

```

zSH> rule show

```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30 auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30 auto-enable-interval (def) 300 600 1200
1/1	bridgeinsertoption82	\$SystemIP\$IfName\$Vlan

3 record(s) found

c Add the bridgeinsertpppoevendortag rule to the downlink bridge.

```

zSH> bridge add 1-6-2-0/eth downlink vlan 888 tagged ipktrule 1
Adding bridge on 1-6-2-0/eth
Created bridge-interface-record 1-6-2-0-eth-888/bridge

```

Applying the filter to this bridge causes the custom strings to be inserted into the packets during the DHCP discovery process.

2 To create a string for the second packetRuleType field:

a To create a string for only the second packetRuleType field of the bridgeinsertpppoevendortag rule:

```

zSH> rule add bridgeinsertoption82 2/1 "" $SystemName
Created packet-rule-record 2/1 (bridgeinsertoption82)

```

b Verify the packet-rule-record.

```

zSH> rule show

```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30 auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30 auto-enable-interval (def) 300 600 1200
1/1	bridgeinsertoption82	\$SystemIP\$IfName\$Vlan
2/1	bridgeinsertoption82	\$SystemName

4 record(s) found

3 To create a rule for the first and the second **packetRuleType** fields:

- a** To create a string for both the first and the second **packetRuleType** fields of the *bridgeinsertppoevendortag* rule:

```
zSH> rule add bridgeinsertoption82 3/1 $SystemName $SystemIP$IfName$Vlan
Created packet-rule-record 3/1 (bridgeinsertoption82)
```

b Verify the **packet-rule-record**.

```
zSH> rule show
```

Group/Member	Type	Value(s)
-----	-----	-----
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30
		auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30
		auto-enable-interval (def) 300 600 1200
1/1	bridgeinsertoption82	\$SystemIP\$IfName\$Vlan
2/1	bridgeinsertoption82	\$SystemName
3/1	bridgeinsertoption82	\$SystemName \$SystemIP\$IfName\$Vlan

5 record(s) found

4 Add the packet rule for *bridgeinsertoption82* to a downlink bridge.

```
zSH> bridge add 1-6-3-0/eth vlan 666 tagged ipktrule 3
Adding bridge on 1-6-3-0/eth
Created bridge-interface-record 1-6-3-0-eth-666/bridge
```

Applying the filter to this bridge causes the custom strings to be inserted into the packets during the DHCP discovery process.

Deleting a packet-rule-record

When necessary, delete the **packet-rule-record**.

Use the **delete packet-rule-record** command.

```
zSH> rule delete 1/1
packet-rule-record 1/1 deleted completely
```

DHCP on bridge packet rules (DHCP relay, and Forbid OUI)

This section describes:

- [DHCP relay, page 323](#)
- [DHCP relay bridge configuration, page 324](#)
- [Forbid OUI, page 327](#)

DHCP relay

Add the DHCP packet rule options using the **rule add** command to specify the packet rule option and which **packet-rule-record** group.

packetRuleValue contains the DHCP subnet group ID. If only the DHCP relay option is used, option82 information is displayed in hex format as *slot port shelf vlan*. See [Configuring bridges to support DHCP relay, page 325](#).

```
zSH> dhcp-relay add 20 11.1.1 NULL
Operation completed successfully.
This DHCP Relay Agent is available only for bridged connections;
Routed interfaces will not be able to use it.
Created DHCP Relay Agent: group: 20, index: 1
```

```
zSH> rule add bridgedhcprelay 10/1 20
Created packet-rule-record 10/1 (bridgedhcprelay)
```

Verify the rule.

```
zSH> rule show
Group/Member                                Type Value(s)
-----
Default dwn (0/1)                          bridgestormdetect discard+alarm+block pps 30 cs 30
                                           auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)                    bridgestormdetect discard+alarm+block pps 100 cs 30
                                           auto-enable-interval (def) 300 600 1200
                                           10/1 bridgedhcprelay 20
3 record(s) found
```

DHCP relay bridge configuration

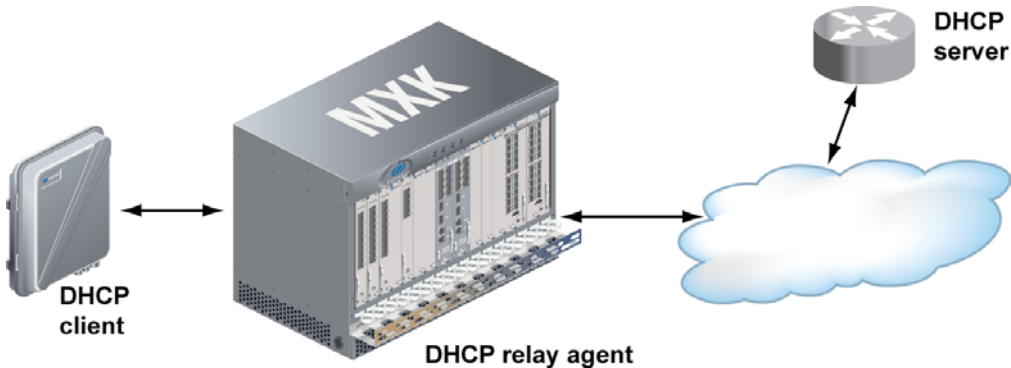
The MXK enables bridges to be configured as DHCP relay agents. All DHCP messages on the bridge will have Option 82 information inserted to be passed up through an IP interface to an external DHCP server.

 **Note:** Bridged DHCP Relay is not supported in IPv6.

The MXK supports primary and alternate DHCP servers, see [IP provisioning examples on page 359](#).

[Figure 44](#) illustrates the traffic flow when the MXK is configured with a bridge to support DHCP relay.

Figure 44: Bridge supported DHCP relay



Configuring bridges to support DHCP relay

This procedure describes how to configure bridges on the MXK to support DHCP relay. You add the DHCP relay as you create the bridge using the **bridge add** command by entering the **dhcp-relay add** command.

Before you add DHCP relay you should have an IP interface on the MXK with a route available to the DHCP server.

After the above elements are configured, use the **dhcp-relay add** command to configure bridge support.

- 1 To configure support for DHCP relay on a bridge use the **dhcp-relay add** command which uses the **subnetgroup** parameter as an identifier:

```
dhcp-relay add [<subnetgroup>] <ip-address> NULL
```

The **subnetgroup** parameter is the index identifier of the dhcp-server subnet.

The **ip-address** parameter is the address of the external DHCP server.

For DHCP relay on bridges you add the **NULL** parameter

- 2 Add the **dhcp-relay** rule using the rule add command which defines that the subnetgroup identifier is in the packet rule group.
- 3 Create bridge (or modify an existing bridge) to include the packet rule group.

Example DHCP relay support on a bridge

- 1 Configure DHCP relay support on the bridge using **dhcp-relay add**.

```
zSH> dhcp-relay add 20 11.1.1.1 NULL
Operation completed successfully.
This DHCP Relay Agent is available only for bridged connections;
Routed interfaces will not be able to use it.
Created DHCP Relay Agent: group: 20, index: 3
```

- 2 Add the dhcp-relay rule to the IP packet rule group.

```
zSH> rule add bridgedhcprelay 10/1 20
Created packet-rule-record 10/1 (bridgedhcprelay)
```

- 3 Create bridge and include IP packet rule group.

```
zSH> bridge add 1-13-10-0/eth downlink vlan 700 ipktrule 10
Adding bridge on 1-13-10-0/eth
Created bridge-interface-record 1-13-10-0-eth/bridge
```

Verify the information in the **dhcp-server-subnet** profile:

```
zSH> get dhcp-server-subnet 3
dhcp-server-subnet 3
network: -----> {0.0.0.0}
netmask: -----> {255.255.255.255}
```

```

domain: -----> {0}
range1-start: -----> {0.0.0.0}
range1-end: -----> {0.0.0.0}
range2-start: -----> {0.0.0.0}
range2-end: -----> {0.0.0.0}
range3-start: -----> {0.0.0.0}
range3-end: -----> {0.0.0.0}
range4-start: -----> {0.0.0.0}
range4-end: -----> {0.0.0.0}
default-lease-time: -----> {-1}
min-lease-time: -----> {-1}
max-lease-time: -----> {-1}
boot-server: -----> {0.0.0.0}
bootfile: -----> {}
default-router: -----> {0.0.0.0}
primary-name-server: ---> {0.0.0.0}
secondary-name-server: -> {0.0.0.0}
domain-name: -----> {}
subnetgroup: -----> {20} dhcp server subnet
stickyaddr: -----> {enable}
external-server: -----> {11.1.1.1} dhcp server address
external-server-alt: ---> {0.0.0.0}

```

- Verify the **dhcp-server-subnet** subnet group.
- Verify the rule exists and the group number:

zSH> **rule show**

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30
		auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30
		auto-enable-interval (def) 300 600 1200
10/1	bridgedhcprelay	20

3 record(s) found

- Verify the **packet-rule-record** links to the DHCP server subnet group:

zSH> **get packet-rule-record 10/1**

```

packet-rule-record 10/1
packetRuleType: ---> {bridgedhcprelay}
packetRuleValue: --> {20}
packetRuleValue2: -> {}
packetRuleValue3: -> {}
packetRuleValue4: -> {}
packetRuleValue5: -> {}

```

- Verify the **bridge-interface-record** contains the packet rule group:

zSH> **get bridge-interface-record 1-13-10-0-eth/bridge**

```

bridge-interface-record 1-13-10-0-eth/bridge
vpi: -----> {0}
vci: -----> {0}

```

```

vlanId: -----> {700}
stripAndInsert: -----> {true}
customARP: -----> {false}
filterBroadcast: -----> {false}
learnIp: -----> {true}
learnUnicast: -----> {true}
maxUnicast: -----> {5}
learnMulticast: -----> {true}
forwardToUnicast: -----> {false}
forwardToMulticast: -----> {false}
forwardToDefault: -----> {true}
bridgeIfCustomDHCP: -----> {false}
bridgeIfIngressPacketRuleGroupIndex: -> {10} packet rule group
vlanIdCOS: -----> {0}
outgoingCOSOption: -----> {disable}
outgoingCOSValue: -----> {0}
s-tagTPID: -----> {0x8100}
s-tagId: -----> {0}
s-tagStripAndInsert: -----> {true}
s-tagOutgoingCOSOption: -----> {s-tagdisable}
s-tagIdCOS: -----> {0}
s-tagOutgoingCOSValue: -----> {0}
mcastControlList: -----> {}
maxVideoStreams: -----> {0}
isPPPoA: -----> {false}
floodUnknown: -----> {false}
floodMulticast: -----> {false}
bridgeIfEgressPacketRuleGroupIndex: --> {0}
bridgeIfTableBasedFilter: -----> {NONE(0)}
bridgeIfDhcpLearn: -----> {NONE(0)}
mvrVlan: -----> {0}
vlan-xlate-from: -----> {0}
slan-xlate-from: -----> {0}

```

Forbid OUI

The *bridgeforbidoui* rule is filtering based on Organizational Unique Identifier (OUI).

When using the *bridgeforbidoui* option for a packet rule, you provide the first three bytes of the MAC address in order to identify the vendor. These three bytes are called the Organizational Unique Identifier (OUI).

```
zSH> rule add bridgeforbidoui 1/1 AA:BB:CC
```

Packets from a device with a MAC address which begins with “AA:BB:CC”, the hexadecimal vendor code, will be blocked.

PPPoE with intermediate agent (bridgeinsertpppoevendortag)

This section covers the two methods used to configure the *bridgeinsertpppoevendortag* rule type for PPPoE Intermediate Agent and includes:

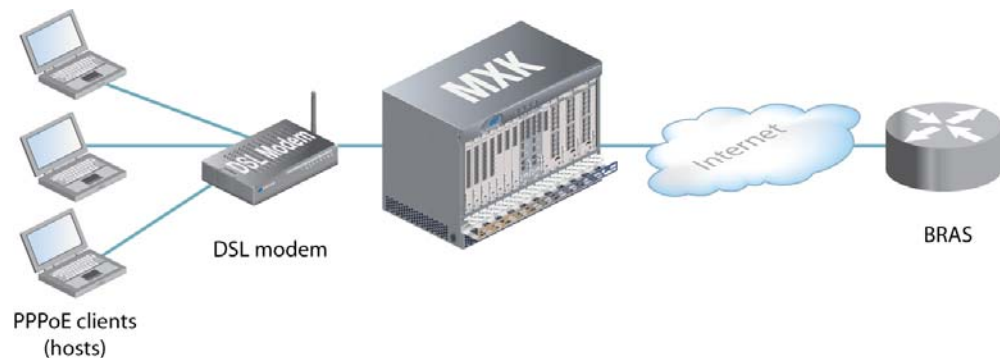
- [PPPoE with intermediate agent overview, page 328](#)
- [PPPoE with intermediate agent configuration without macro defined strings, page 329](#)
- [PPPoE with intermediate agent configuration with macro defined strings, page 331](#)

PPPoE with intermediate agent overview

PPP headend servers (also known as Broadband Remote Access Servers or BRAS) authenticate and manage PPP connections.

TR-101 defines information which is entered into the packets when creating Point-to-Point Protocol over Ethernet connection through an Intermediate Agent (PPPoE Intermediate Agent).

Figure 45: PPPoE with intermediate agent



The MXK is capable of being an intermediate agent in a PPPoE (point-to-point protocol over Ethernet) scenario as shown in [Figure 45](#).

In a PPPoE scenario, PPPoE clients initiate the connection process and need to learn the Ethernet address of the remote peer and establish a unique session ID to establish a connection.

PADI

During the discovery process, the PPPoE client (host) broadcasts a request by transmitting PPPoE Active Discovery Initiation (PADI) packets. When one or more responses are received by the host (the responses include the address of the Broadband Remote Access Server (BRAS)), the host then sends a unicast PPPoE Active Discovery Request (PADR) packet.

PADS

The MXK automatically inserts slot, port, SLAN/VLAN information into PPPoE packets that transits a MXK bridge interface. The MXK can also be configured to insert a customized string into the vendor-specific portion of the PPPoE packet when receiving a PPPoE Active Discovery Initiation (PADI) packet or a PPPoE Active Discovery Request (PADR) packet.

The customized string is entered into the **packetRuleValue** field of the **rule add** command.

The MXK supports two ways to configure the packetRuleValue for the *bridgeinsertpppoevendortag* rule type. The first is without macro defined strings, see [PPPoE with intermediate agent configuration without macro defined strings on page 329](#). The second is with macro defined strings, see [PPPoE with intermediate agent configuration with macro defined strings on page 331](#).

Without macro defined strings, PPPoE behavior prepends as much text of the custom string as will fit in the field (from 0 to 48 characters) and the output text is truncated if required to fit into the **packetRuleValue** field.

PPPoE with intermediate agent configuration without macro defined strings

The customized identification string is 0 to 48 characters. The inserted information is TR-101 compliant and formatted as:

<customstring> eth slot/port [[:stagID]:vlan-tag] slot/port SLAN and VLAN is default information automatically inserted into the packet

The structure of the rule is that if a custom string is entered, that string, and only that string, will be entered in the packet. If a custom string is not entered, the eth slot/port [[:stagID]:vlan-tag] is entered.

The *slot/port* identifies the ingress *slot/port* on the MXK where the packet was received. If the bridge is configured with a VLAN or SLAN tag, the VLAN/SLAN tag is also added to the packet.

When the **packetRuleValue** field is blank or contains a text string without a dollar sign, the **packetRuleValue** field is processed as shown in [Creating a packet rule for bridgeinsertpppoevendortag for default information on page 329](#).

Creating a packet rule for bridgeinsertpppoevendortag for default information

Create a **packet-rule-record** with for default information.

- 1 Create the *bridgeinsertpppoevendortag* filter for default information.

```
zSH> rule add bridgeinsertpppoevendortag 1/1 ""
Created packet-rule-record 1/1 (bridgeinsertpppoevendortag)
```

- 2 Verify the **packet-rule-record**.

```
zSH> rule show
```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30 auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30 auto-enable-interval (def) 300 600 1200

```
1/1      bridgeinsertpppoevendortag
3 record(s) found
```

3 Add the packet rule to a downlink bridge.

```
zSH> bridge add 1-6-1-0/eth downlink vlan 101 tagged ipktrule 1
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-101/bridge
```

Applying the filter to this bridge causes the eth *slot/port* *[:stagID]:vlan-tag* to be inserted into the packets for PPPoE session establishment.



Note: For configurations with bridge intralinks or subtended SLMS devices, ensure that the PPPoE intermediate agent feature is enabled on only the subtended devices, or the downlink, or the TLS bridges.

Creating a packet rule for bridgeinsertpppoevendortag rule with custom string

1 Enter the **rule add** command with group/member index and custom string.

```
zSH> rule add bridgeinsertpppoevendortag 2/1 test_mxk
Created packet-rule-record 2/1 (bridgeinsertpppoevendortag)
```

2 Verify the rule.

```
zSH> rule show
Group/Member                                     Type Value(s)
-----
Default dwn (0/1)                               bridgestormdetect discard+alarm+block pps 30 cs 30
                                              auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)                         bridgestormdetect discard+alarm+block pps 100 cs 30
                                              auto-enable-interval (def) 300 600 1200
1/1      bridgeinsertpppoevendortag
2/1      bridgeinsertpppoevendortag test_mxk
4 record(s) found
```

3 Apply the packet rule to a downlink bridge.

```
zSH> bridge add 1-6-2-0/eth downlink vlan 201 tagged ipktrule 2
Adding bridge on 1-6-2-0/eth
Created bridge-interface-record 1-6-2-0-eth-201/bridge
```

Applying the filter to this bridge causes the custom string *test_mxk* to be inserted into the packets for PPPoE session establishment.

Deleting a packet-rule-record

When necessary, delete the **packet-rule-record**.

Use the **delete packet-rule-record** command.

```
zSH> rule delete 1/1
packet-rule-record 1/1 deleted completely
```

PPPoE with intermediate agent configuration with macro defined strings

The MXK can be configured to insert a customized string into the vendor-specific portion of the PPPoE packet when receiving a PPPoE Active Discovery Initiation (PADI) packet or a PPPoE Active Discovery Request (PADR) packet.

If the **packetRuleValue** field contains one or more dollar sign (\$) characters, the vendor tag text that would have been supplied is replaced by the contents of the field as follows:

This section discusses how to insert customized strings with the use of supported macro formats as shown in [Table 22](#).

- When a dollar sign character is encountered, the text following the dollar sign is compared to [Table 22](#).
- If no match is found the dollar sign character is replaced with the text "Unknown".
- If a match is found the dollar sign character and the associated text is replaced by the text indicated.
- The macro name and abbreviations are both case sensitive.
- The \$macro strings may be imbedded in literal text. This text is copied to the output without change.
- The supported macro formats may be entered in the text as either \$macroname or \$abbreviation. Thus \$SystemName and \$NM give the same result, which is to substitute the system name from the **system 0** profile.

Some of the macros vary in effect depending on the value they are intended to display.

- \$Gem and \$Onu IDs are displayed or not depending on whether or not they have a non-zero value.
- \$Vlan displays -SLAN-VLAN if the SLAN is non-zero, -VLAN if the -SLAN is zero but the VLAN is non-zero, or nothing if they are both zero.
- \$VC displays -vpi-vci if either value is non-zero and nothing if they are both zero.



Note: Macro names are case sensitive.

Table 22: Supported macro formats for macro defined strings

Macro name	Abbreviation	Varies	Result
\$SystemName	NM	NM	sysname from the system 0 profile.
\$SystemIP	IP	No	ipaddress address from the system 0 profile.
\$IfName	IF	IF	ifName from the bridge IfTranslate profile.
\$Address	AD	No	<i>shelf-slot-port-subport-type</i> of the underlying physical interface. Where the interface is a GPON OLT interface the type is given as <i>gponport</i> and the subport is the GEM port.
\$Shelf	SH	No	Shelf (currently always 1).
\$Slot	SL	No	<i>slot</i> from the IfTranslate profile of the underlying physical interface.
\$Port	PT	No	<i>port</i> (see \$Slot).
\$SubPort	SP	No	<i>subport</i> (see \$Slot.) For GPON this is the GEM port
\$Gem	GM	Yes	-GEMPort (or nothing)
\$Onu	ON	Yes	-ONUnumber (or nothing)
\$Type	TY	No	Type (for GPON this is <i>gponport</i>).
\$Vlan	VN	Yes	-SLAN-VLAN (or -VLAN or nothing).
\$Svlan	SV	No	SLAN
\$Cvlan	CV	No	VLAN
\$Vc	VC	Yes	-VPI-VCI (or nothing)
\$Vpi	VP	No	-VPI
\$Vci	VI	No	-VCI
\$Null	NL	No	Nothing (used to change PPPoE handling of constant strings).

Creating a packet rule for bridgeinsertpppoevendortag using macro names

Create a **packet-rule-record** using macro names to create a user-defined string. Strings created with macros, including the information pulled in by the macro, are limited to 48 characters.

- 1 To create a string with macro names that includes *shelf/slot/port/subport*, VLAN ID, and SLAN ID enter:

```
zSH> rule add bridgeinsertpppoevendortag 3/1
$SystemName$Shelf$Slot$Port$Subport$Vlan$Svlan
Created packet-rule-record 3/1 (bridgeinsertpppoevendortag)
```


2 Verify the packet-rule-record.

```
zSH> rule show
-----
Group/Member                                Type Value(s)
-----
Default dwn (0/1)                          bridgestormdetect discard+alarm+block pps 30 cs 30
                                           auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)                     bridgestormdetect discard+alarm+block pps 100 cs 30
                                           auto-enable-interval (def) 300 600 1200
1/1    bridgeinsertpppoevendortag
2/1    bridgeinsertpppoevendortag test_mvk
3/1    bridgeinsertpppoevendortag
$SystemName$Shelf$Slot$Port$Subport$Vlan$Svlan
5 record(s) found
```

3 Apply the bridgeinsertpppoevendortag rule to the downlink bridge.

```
zSH> bridge add 1-6-3-0/eth downlink vlan 301 tagged ipktrule 3
Adding bridge on 1-6-3-0/eth
Created bridge-interface-record 1-6-3-0-eth-301/bridge
```

The ifName (typically *shelf/slot/port/subport*, and the VLAN ID and SLAN ID configured on the bridge will be inserted into the packets for PPPoE session establishment.

Deleting a packet-rule-record

When necessary, delete the **packet-rule-record**.

Use the **delete packet-rule-record** command.

```
zSH> rule delete 3/1
packet-rule-record 3/1 deleted completely
```

Creating a packet rule for bridgeinsertpppoevendortag rule using macro names and text

You can create a *bridgeinsertpppoevendortag* filter that combines macro names and text.

- 1 To create a string with macro names and text, in this case oakland and system name, enter

```
zSH> rule add bridgeinsertpppoevendortag 4/1 oakland$IfName$Vlan$Svlan
Created packet-rule-record 4/1 (bridgeinsertpppoevendortag)
```

2 Verify the packet-rule-record.

```
zSH> rule show
-----
Group/Member                                Type Value(s)
-----
Default dwn (0/1)                          bridgestormdetect discard+alarm+block pps 30 cs 30
                                           auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)                     bridgestormdetect discard+alarm+block pps 100 cs 30
                                           auto-enable-interval (def) 300 600 1200
1/1    bridgeinsertpppoevendortag
```

```

2/1      bridgeinsertpppoevendortag test_mvk
3/1      bridgeinsertpppoevendortag
$SystemName$Shelf$Slot$Port$Subport$Vlan$Svlan
4/1      bridgeinsertpppoevendortag oakland$IfName$Vlan$Svlan
6 record(s) found

```

3 Apply the packet rule to the downlink bridge.

```

zSH> bridge add 1-6-4-0/eth downlink vlan 401 tagged ipktrule 4
Adding bridge on 1-6-4-0/eth
Created bridge-interface-record 1-6-4-0-eth-401/bridge

```

Applying the filter to this bridge causes the custom string to be inserted into the packets for PPPoE session establishment.

Deleting a packet-rule-record

When necessary, delete the **packet-rule-record**.

Use the **delete packet-rule-record** command.

```

zSH> rule delete 4/1
packet-rule-record 4/1 deleted completely

```

Bandwidth limiting by port and service, single and dual rate limiting

This section describes these topics:

- [Rate limiting overview, page 334](#)
- [Configure color blind rate limiting, page 337](#)
- [Configure color blind policing single rate, page 339](#)
- [Color to Cos default values, page 347](#)
- [Configure color aware rate limiting, page 343](#)
- [DSCP to COS \(802.1p\) mapping, page 348](#)

Rate limiting overview

Rate limiting on the MXK allows for two types of policing for both color blind and color aware rate limiting:

- Single rate (CIR)
- Dual Rate (CIR and PIR)

Single rate, allows service providers to provide customers services with limited bandwidth with the Committed Information Rate (CIR) as the committed rate and all traffic up to the CIR is guaranteed.

Dual rate limiting allows service providers to configure rate limitations on a per VLAN basis to limit traffic based on two rates, the CIR and the Peak Information Rate (PIR). In this case all traffic up to the CIR is guaranteed and

all traffic above the PIR is discarded. Traffic between the CIR and the PIR is handled on a best effort basis.

After configuring an interface with rate limiting, the traffic rate is monitored and metered to verify conformity with an established contract. Non-conforming traffic is discarded, while conforming traffic passes through the interface without any changes. The MXK follows RFC 2697 for rate limiting on both the ingress and egress of the interface.

The modes of rate limiting on the MXK are:

- Single and dual rate limiting color blind

Rate limiting is performed on the interface without using the frame's Class of Service (CoS) by assuming that all packets of a flow are “uncolored” and are treated equally when in the range of the CIR.

For color blind dual rate limiting, packets are treated equally up to the CIR and are treated on a best effort basis between the rates set by the CIR and the PIR.

You can configure yellow markings on a single and dual rate packet rules. In this case, a CoS value is inserted into packets that exceed the CIR.

Color blind mode is most commonly used for a single service per VLAN.

- Single and dual rate limiting color aware

Rate limiting observes that the incoming packet flow is colored and each packet is marked green, yellow, or red to signify if a packet has high, medium, or low priority. The color field maps to the priority CoS value in tagged packets and the IP precedence ToS value in untagged packets.

Color aware mode is most commonly used for multiple services on a single VLAN to ensure that the higher priority packets get through if there is bandwidth contention.

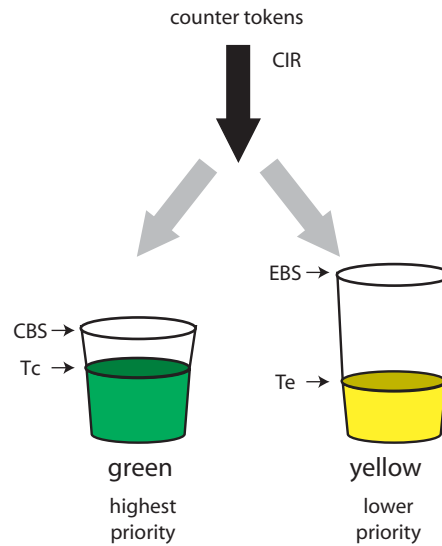


Note: Color values are not supported on egress ports.

Single rate counter scheme

The single rate color marker scheme from RFC 2697 uses a counter to gauge capacity on the line by counting tokens. The counters are used to determine which packets get dropped. The idea is that the green bucket fills up faster than the yellow buckets.

There are three parameters which determine which packets are dropped — a CIR which supplies tokens to be counted, and two buckets, Committed Burst Size (CBS) and Excess Burst Size (EBS), which provide two levels of priority. [Figure 46](#) describes a single rate counter scheme.

Figure 46: Single rate counter scheme

CIR is the rate which determines how quickly the token buckets fill up. Both buckets start full. It is important to understand that this is not a buffering scheme as incoming packets are not queued up for later delivery.

For every CIR increment the buckets are filled.

```

if Tc < CBS
  then
    increment Tc
  else if Te < EBS
    then
      increment Te
    else
      do nothing (do not increment either because
                  they are both full)

```

The green bucket will fill first and faster if it is not full because the yellow bucket will not increment until $Tc \geq CBS$.

There are rules about how the green bucket size (CBS) and yellow bucket size (EBS) should be configured. At least one of CBS or EBS should be greater than zero. Also at least one of CBS or EBS should be greater than the largest expected packet in the incoming stream, as packets which are larger than both CBS or EBS will be dropped. Normally you would have CBS greater than EBS, so packets that do not go because there are not enough green tokens will go because there are enough yellow tokens.

With color blind rate limiting the size of the incoming packet determines whether the packet will go. If there are enough tokens in green or yellow it will go. Tokens matching the size of the packet will be decremented from the appropriate bucket. If there are packets which are larger than the amount of tokens in either bucket, those packets are dropped. Packets which are larger than either bucket size when full are dropped.

```

if incoming packet smaller than Tc

```

```

then
    decrement Tc by size of packet
    send packet
else if packet smaller than Te
    then
        deccrement Te by size of packet
        send packet
    else
        drop packet

```

With color aware rate limiting, it is assumed that the packets are being marked by an upstream device. Packets which are marked red are dropped. Packets which are marked yellow are best effort and green are highest priority and should have the lowest chance of the packet being dropped. The behavior depends on the configuring of the CBS and EBS parameters.



Note: The default values for CBS and EBS are good for most situations. Only advanced users should change these values.

With color aware rate limiting the size and the color determine whether the packet will be dropped.

```

if incoming packet is green AND is smaller than Tc
    then
        decrement Tc by size of packet
        send packet
    else if packet is green or yellow AND is smaller than
Te
        then
            deccrement Te by size of packet
            send packet
        else
            drop packet

```

So all red packets are dropped. Normally the upstream marking device will mark packets red which are too large.

Configure color blind rate limiting

This section describes single and dual color blind rate limiting and includes:

- [Rate limiting controls, page 338](#)
- [Configure color blind policing single rate, page 339](#)
- [Configure color blind policing dual rate, page 342](#)

Color blind rate limiting is usually set when one service is supplied per VLAN. The committed information rate, CIR, is set in kilobytes per second. For any rate above the set CIR, packets will drop.

For example, in [Figure 47](#), you would use the color blind method to set VLAN 100 to drop packets when the rate exceeds 5 Mbps, VLAN 200 to drop

packets when the rate exceeds 3 Mbps, and VLAN 200 to drop packets when the rate exceeds 6 Mbps.

Figure 47: One service per VLAN on an interface



Rate limiting controls

The syntax for color blind rate limiting is:

```
rule add ratelimitdiscard <groupIndex/memberIndex> rate <rate> [peak <value>] [cbs <value>] [ebs <value>] [ymark <value>]
```

Table 23: Definition of rate limiting controls

Acronym	Definition	Rate	Description
rate	Committed Information Rate (CIR)	kbps	The average rate guaranteed for a virtual circuit. If the actual rate goes above the CIR the packets will be dropped.
peak rate	Peak Information Rate (PIR)	kbps	The peak rate in which traffic above this rate is discarded and traffic between the CIR and PIR is handled on a best effort basis.
cbs	Committed Burst Size	bps	The maximum data rate which can be carried under normal conditions. This rate is greater than the CIR, but less than the EBS.
ebs	Excess Burst Size	bps	The maximum data rate that the circuit will attempt to carry.
ymark	yellow marking		Packets are marked with the provided value. When the parameter is not configured, yellow packets are untouched. The range is 0-7.

 **Note:** The default values for CBS and EBS are good for most situations. Only advanced users should change these values.

Configure color blind policing single rate

The **rule add ratelimitdiscard** command sets the rate above which packets will be dropped for single rate limiting.

```
rule add ratelimitdiscard <groupIndex/memberIndex> rate <rate> [peak <value>] [cbs
<value>] [ebs <value>] [ymark <value>]
```

Dual rate limiting is allowed on the egress only.

Color blind policing works on both the ingress and egress for single rate limiting.

Case 1: Configure a color blind policing filter for the ingress of a bridge for single rate limiting

This example describes setting the rate above which packets are dropped on an subscriber facing GPON bridge, in this case on the ingress of the bridge.

- 1 Create the packet rule for the ingress from the subscriber to the MXK.

```
zSH> rule add ratelimitdiscard 1/1 rate 1800
Created packet-rule-record 1/1 (ratelimitdiscard)
```

- 2 Verify the rule.

```
zSH> rule show
```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30
		auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30
		auto-enable-interval (def) 300 600 1200
1/1	ratelimitdiscard	cir 1800kbps cbs 120000bytes ebs

```
130000bytes
3 record(s) found
```

To view packet rules by type, enter the rule type, **ratelimitdiscard**:

```
zSH> rule show ratelimitdiscard
```

Group/Member	Type	Value(s)
1/1	ratelimitdiscard	cir 1800kbps cbs 120000bytes ebs

```
130000bytes
1 record(s) found
```

- 3 Apply the rule to the ingress of the Ethernet MXK bridge.

```
zSH> bridge add 1-6-1-0/eth downlink vlan 777 ipktrule tagged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth/bridge
```

- 4 Create a network facing uplink bridge with VLAN 777.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 777 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-777/bridge
```

Bridge-path added successfully

Case 2: Configuring color blind policing filters for both the ingress and the egress of a bridge for single rate limiting

This example describes how service providers can use two color blink rate limiting filters to service subscribers that will allow higher bandwidth coming from the network through the MXK to the subscriber and lower bandwidth leaving the subscriber through the MXK to the network on single rate limiting.

1 Create the packet rule for the ingress from the subscriber to the MXK.

```
zSH> rule add ratelimitdiscard 2/1 rate 1300
Created packet-rule-record 2/1 (ratelimitdiscard)
```

2 Create the rule for the egress from the MXK to the subscriber.

```
zSH> rule add ratelimitdiscard 3/1 rate 6000
Created packet-rule-record 3/1 (ratelimitdiscard)
```

3 View the rules.

```
zSH> rule show
Group/Member                                Type Value(s)
-----
Default dwn (0/1)                          bridgestormdetect discard+alarm+block pps 30 cs 30
                                           auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)                     bridgestormdetect discard+alarm+block pps 100 cs 30
                                           auto-enable-interval (def) 300 600 1200
130000bytes                                1/1                                ratelimitdiscard cir 1800kbps cbs 120000bytes ebs
130000bytes                                2/1                                ratelimitdiscard cir 1300kbps cbs 120000bytes ebs
130000bytes                                3/1                                ratelimitdiscard cir 6000kbps cbs 120000bytes ebs
5 record(s) found
```

To view just the **ratelimitdiscard** rules enter:

```
zSH> rule show ratelimitdiscard
Group/Member                                Type Value(s)
-----
1/1                                ratelimitdiscard cir 1800kbps cbs 120000bytes ebs 130000bytes
2/1                                ratelimitdiscard cir 1300kbps cbs 120000bytes ebs 130000bytes
3/1                                ratelimitdiscard cir 6000kbps cbs 120000bytes ebs 130000bytes
3 record(s) found
```

4 Apply the rules to both the ingress and the egress of the Ethernet MXK bridge.

```
zSH> bridge add 1-6-1-0/eth downlink vlan 888 ipktrule 2 epktrule 3 tagged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-888/bridge
```




Note: Both packet rules must be applied to the bridge interface with the same **bridge add** command.

5 Verify the packet rules.

```
zSH> get bridge-interface-record 1-6-1-0-eth-888/bridge
bridge-interface-record 1-6-1-0-eth-888/bridge
vpi: -----> {0}
vci: -----> {0}
vlanId: -----> {888}
stripAndInsert: -----> {false}
customARP: -----> {false}
filterBroadcast: -----> {false}
learnIp: -----> {true}
learnUnicast: -----> {true}
maxUnicast: -----> {5}
learnMulticast: -----> {true}
forwardToUnicast: -----> {false}
forwardToMulticast: -----> {false}
forwardToDefault: -----> {true}
bridgeIfCustomDHCP: -----> {false}
bridgeIfIngressPacketRuleGroupIndex: --> {2}
vlanIdCOS: -----> {0}
outgoingCOSOption: -----> {disable}
outgoingCOSValue: -----> {0}
s-tagTPID: -----> {0x8100}
s-tagId: -----> {0}
s-tagStripAndInsert: -----> {true}
s-tagOutgoingCOSOption: -----> {s-tagdisable}
s-tagIdCOS: -----> {0}
s-tagOutgoingCOSValue: -----> {0}
mcastControlList: -----> {}
maxVideoStreams: -----> {0}
isPPPoA: -----> {false}
floodUnknown: -----> {false}
floodMulticast: -----> {false}
bridgeIfEgressPacketRuleGroupIndex: --> {3}
bridgeIfTableBasedFilter: -----> {NONE(0)}
bridgeIfDhcpLearn: -----> {NONE(0)}
mvrVlan: -----> {0}
vlan-xlate-from: -----> {0}
slan-xlate-from: -----> {0}
bridge-type: -----> {downlink}
```

6 Create a network facing uplink bridge with VLAN 888.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 888 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-888/bridge
Bridge-path added successfully
```

Configure color blind policing dual rate

The **rule add ratelimitdiscard** command sets the range for the committed rate (CIR) and the peak rate (PIR). Packets above the PIR will be discarded and traffic between the CIR and the PIR will be handled on a best effort basis.

Case 3: Configuring a color blind policing filter for dual rate limiting

This example describes setting the committed rate (CIR) and the peak rate (PIR) above which packets are dropped on an subscriber facing GPON bridge. Packets between the CIR and PIR will be handled on a best effort basis.



Note: Dual color blind policing works only on the egress for dual rate limiting.

- 1 Create the dual rate limiting rule to apply to the egress of the Ethernet downlink bridge.

```
zSH> rule add ratelimitdiscard 4/1 rate 2000 peak 4000
Created packet-rule-record 4/1 (ratelimitdiscard)
```

- 2 Verify the rule.

```
zSH> rule show
```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30
		auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30
		auto-enable-interval (def) 300 600 1200
1/1	ratelimitdiscard	cir 1800kbps cbs 120000bytes ebs
130000bytes		
2/1	ratelimitdiscard	cir 1300kbps cbs 120000bytes ebs
130000bytes		
3/1	ratelimitdiscard	cir 6000kbps cbs 120000bytes ebs
130000bytes		
4/1	ratelimitdiscard	cir 2000kbps cbs 120000bytes pir 4000kbps
ebs 130000bytes		
6 record(s) found		

- 3 Apply the rule to the egress of the Ethernet downlink bridge.

```
zSH> bridge add 1-6-1-0/eth downlink vlan 999 epktrule 4 tagged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-999/bridge
```

- 4 Create a network facing uplink bridge with VLAN 999.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 999 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-999/bridge
Bridge-path added successfully
```

Case 4: Configuring a color blind policing filter for dual rate limiting with ymark

This example describes setting the committed rate (CIR) and the peak rate (PIR) above which packets are dropped on an subscriber facing GPON bridge. Packets between the CIR and PIR will be handled on a best effort basis.

You can use the **ymark** value to mark packets that flow between the CIR and the PIR for color aware upstream network devices.



Note: Dual color blind policing works only on the egress for dual rate limiting.

- 1 Create the dual rate limiting rule to apply to the egress of the GPON downlink bridge.

```
zSH> rule add ratelimitdiscard 3/1 rate 18000 peak 36000 ymark 1
Created packet-rule-record 3/1 (ratelimitdiscard)
```

- 2 View the rules.

```
zSH> rule show
```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30 auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30 auto-enable-interval (def) 300 600 1200
3/1	ratelimitdiscard	cir 18000kbps cbs 400000bytes pir 36000kbps ebs 400000bytes ym 1

3 record(s) found

- 3 Apply the rule to the egress of the GPON downlink bridge.

```
zSH> bridge add 1-6-1-0/eth downlink vlan 666 epktrule 3 tagged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-666/bridge
```

- 4 Create a network facing uplink bridge with VLAN 666.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 666 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-666/bridge
Bridge-path added successfully
```

Configure color aware rate limiting

This section describes single and dual color aware rate limiting and includes:

- [Rate limiting controls, page 344](#)
- [Configure color aware policing single rate, page 345](#)
- [Configure color aware policing dual rate, page 345](#)

Color aware rate limiting is usually set when more than one service is supplied per VLAN.

The high-priority and low-priority parameters allow you to designate which values on the scale will be treated as green, yellow and red. If high-priority is set to 5 and the low-priority set to 3, the CoS value to color table will change so that 7, 6, and 5 are green; 4 and 3 will be yellow; and 2, 1 and 0 will be dropped.

Rate limiting controls

The syntax for color blind rate limiting is:

```
rule add colorawareratelimitdiscard <groupIndex/memberIndex> rate <rate> [peak
<value>] [cbs <value>] [ebs <value>] [ymark <value>] [hi-priority <value>]
[low-priority <value>]
```

Table 24: Definition of color aware rate limiting controls

Acronym	Definition	Rate	Description
rate	Committed Information Rate (CIR)	kbps	The average rate guaranteed for a virtual circuit. If the actual rate goes above the CIR the packets will be dropped.
peak rate	Peak Information Rate (PIR)	kbps	The peak rate in which traffic above this rate is discarded and traffic between the CIR and PIR is handled on a best effort basis.
cbs	Committed Burst Size	bps	The maximum data rate which can be carried under normal conditions. This rate is greater than the CIR, but less than the EBS.
ebs	Excess Burst Size	bps	The maximum data rate that the circuit will attempt to carry.
ymark	yellow marking		Packets are marked with the provided value, when the parameter is not configured, yellow packets are untouched. The range is 0-7.
hi	hi-priority		Packets are marked according to the colors that correspond to CoS values. See Figure 25 .
lo	low-priority		Packets are marked according to the colors that correspond to CoS values. See Figure 25 .



Note: The default values for CBS and EBS are good for most situations and are set according to device. Only advanced users should change these values.

Configure color aware policing single rate

The **rule add colorawareratelimitdiscard** command sets the color priority and the rate above which packets will be dropped.

```
rule add colorawareratelimitdiscard <groupIndex/memberIndex> rate <rate> [peak
<value>] [cbs <value>] [ebs <value>] [ymark <value>] [hi-priority <value>]
[low-priority <value>]
```

Case 1: Configuring color aware policing filters for the egress of a bridge for single rate

- 1 Create the color aware rule for the egress.

```
zSH> rule add colorawareratelimitdiscard 1/1 rate 1300
Created packet-rule-record 1/1 (colorawareratelimitdiscard)
```

The hi-priority and low-priority are set at the defaults as shown in [Table 25](#).

- 2 View the rule.

```
zSH> rule show
```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30 auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30 auto-enable-interval (def) 300 600 1200
1/1	colorawareratelimitdiscard	cir 1300kbps cbs 120000bytes ebs 130000bytes hi 4 lo 0

3 record(s) found

- 3 Apply the rule for the egress on the Ethernet MXK bridge.

```
zSH> bridge add 1-6-1-0/eth downlink vlan 555 epktrule 1 tagged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-555/bridge
```

- 4 Create a network facing uplink bridge with VLAN 555.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 555 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-555/bridge
Bridge-path added successfully
```

Configure color aware policing dual rate

The **rule add colorawareratelimitdiscard** command sets the range for the committed rate (CIR) and the peak rate (PIR). Packets above the PIR will be

discarded and traffic between the CIR and the PIR will be handled on a best effort basis.

Case 2: Configuring a color blind policing filter for dual rate limiting

This example describes setting the committed rate (CIR) and the peak rate (PIR) above which packets are dropped on the egress of a subscriber facing GPON bridge. Packets between the CIR and PIR will be handled on a best effort basis.



Note: Dual color aware policing works only on the egress.

- 1 Create the color aware dual rate limiting rule for the egress.

```
zSH> rule add colorawareratelimitdiscard 2/1 rate 1800 peak 3600
Created packet-rule-record 2/1 (colorawareratelimitdiscard)
```

- 2 View the rule.

```
zSH> rule show
Group/Member                                     Type Value(s)
-----
Default dwn (0/1)                               bridgestormdetect discard+alarm+block pps 30 cs 30
                                                auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)                           bridgestormdetect discard+alarm+block pps 100 cs 30
                                                auto-enable-interval (def) 300 600 1200
1/1 colorawareratelimitdiscard cir 1300kbps cbs 120000bytes ebs
130000bytes hi 4 lo 0
2/1 colorawareratelimitdiscard cir 1800kbps cbs 120000bytes pir 3600kbps
ebs 130000bytes hi 4 lo 0
4 record(s) found
```

- 3 Apply the rule for the egress on the Ethernet MXK bridge.

```
zSH> bridge add 1-6-1-0/eth downlink vlan 444 epktrule 2 tagged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-444/bridge
```

- 4 Create a network facing uplink bridge with VLAN 444.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 444 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-444/bridge
Bridge-path added successfully
```

Case 3: Configuring a color blind policing filter for dual rate limiting with ymark

This example describes setting the committed rate (CIR) and the peak rate (PIR) above which packets are dropped on the egress of a subscriber facing

GPON bridge. Packets between the CIR and PIR will be handled on a best effort basis.

You can use the **ymark** value to mark packets that flow between the CIR and the PIR for color aware upstream network devices.



Note: Dual color aware policing works only on the egress.

1 Create the color aware dual rate limiting rule for the egress.

```
zSH> rule add colorawareratelimitdiscard 3/1 rate 1800 peak 3600 ymark 1
Created packet-rule-record 3/1 (colorawareratelimitdiscard)
```

2 View the rule.

```
zSH> rule show
```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30
		auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30
		auto-enable-interval (def) 300 600 1200
3/1	colorawareratelimitdiscard	cir 1800kbps cbs 120000bytes pir 3600kbps
ebs 130000bytes ym 1 hi 4 lo 0		

3 record(s) found

3 Apply the rule for the egress on the Ethernet MXK bridge.

```
zSH> bridge add 1-6-1-0/eth downlink vlan 333 ipktrule 3 tagged
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-333/bridge
```

4 Create a network facing uplink bridge with VLAN 400.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 333 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-333/bridge
Bridge-path added successfully
```

Color to Cos default values



Note: Not commonly used except when performing advanced configurations.

Color aware bandwidth limiting is usually used when multiple services with different priorities are offered on a single VLAN. The colors green, yellow, and red are used for metering traffic and the colors correspond to CoS values that range from 0-7. You can set which colors correspond to which CoS value.

Color Aware Policing is based on the idea that upstream devices are policing and marking frames based on a set of rules. A green packet is well behaved. A yellow packet has misbehaved at some point so if there is a bandwidth congestion it should be dropped before a green frame. A red packet has

violated a rule and should be dropped. This means that green packets are serviced first, then if there is enough room, the yellow packets are serviced. Red packets are always dropped.

Table 25 shows the default mapping of CoS value to color.

Table 25: Default Color to CoS values

CoS value	Color
7	green
6	green
5	green
4	green
3	yellow
2	yellow
1	yellow
0	yellow

DSCP to COS (802.1p) mapping



Note: DSCP to COS (802.1p) is available on GPON.

Some network architectures require QoS prioritization at layer 2 and others at layer 3. In order to maintain QoS between Layer 2 Ethernet and Layer 3 IP protocols, the MXK now supports mapping Differentiated Services Code Points (DSCP) to Classes of Services (CoS) as defined by IEEE 802.1p.

CoS a layer 2 QoS marking mechanism involves manipulating the layer 2 Ethernet 802.1p tag. CoS uses 3 bits and therefore values can be anything from 0 to 7. DSCP involves manipulating the IP header info (specifically the ToS field). DSCP uses 6 bits and value range from 0 to 63. DSCP and ToS are different use of the same bits. Therefore, the following standard mapping table can be used as a reference when provisioning DSCP to COS (802.1p).

Table 26: Default DSCP to CoS (802.1p) mapping

DSCP	0–7	8–15	16–23	24–31	32–39	40–47	48–55	56–63
CoS	0	1	2	3	4	5	6	7

Creating a packet-rule-record for DSCP to CoS for bridges

You can create a **packet-rule-record** for DSCP to CoS for new or existing bridges, usually on the network facing Ethernet port (ingress).

- 1 View the mapping in the **dscp-to-cos** profile.


```

zSH> get dscp-to-cos 1
dscp-to-cos 1
dscp00map8021p: -> {0}
dscp01map8021p: -> {0}
dscp02map8021p: -> {0}
dscp03map8021p: -> {0}
dscp04map8021p: -> {0}
dscp05map8021p: -> {0}
dscp06map8021p: -> {0}
dscp07map8021p: -> {0}
dscp08map8021p: -> {1}
dscp09map8021p: -> {1}
dscp10map8021p: -> {1}
dscp11map8021p: -> {1}
dscp12map8021p: -> {1}
dscp13map8021p: -> {1}
dscp14map8021p: -> {1}
dscp15map8021p: -> {1}
dscp16map8021p: -> {2}
dscp17map8021p: -> {2}
dscp18map8021p: -> {2}
dscp19map8021p: -> {2}
dscp20map8021p: -> {2}
dscp21map8021p: -> {2}
dscp22map8021p: -> {2}
dscp23map8021p: -> {2}
dscp24map8021p: -> {3}
dscp25map8021p: -> {3}
dscp26map8021p: -> {3}
dscp27map8021p: -> {3}
dscp28map8021p: -> {3}
dscp29map8021p: -> {3}
dscp30map8021p: -> {3}
dscp31map8021p: -> {3}
dscp32map8021p: -> {4}
dscp33map8021p: -> {4}
dscp34map8021p: -> {4}
dscp35map8021p: -> {4}
dscp36map8021p: -> {4}
dscp37map8021p: -> {4}
dscp38map8021p: -> {4}
dscp39map8021p: -> {4}
dscp40map8021p: -> {5}
dscp41map8021p: -> {5}
dscp42map8021p: -> {5}
dscp43map8021p: -> {5}
dscp44map8021p: -> {5}
dscp45map8021p: -> {5}
dscp46map8021p: -> {5}
dscp47map8021p: -> {5}
dscp48map8021p: -> {6}
dscp49map8021p: -> {6}
dscp50map8021p: -> {6}
dscp51map8021p: -> {6}

```

```

dscp52map8021p: -> {6}
dscp53map8021p: -> {6}
dscp54map8021p: -> {6}
dscp55map8021p: -> {6}
dscp56map8021p: -> {7}
dscp57map8021p: -> {7}
dscp58map8021p: -> {7}
dscp59map8021p: -> {7}
dscp60map8021p: -> {7}
dscp61map8021p: -> {7}
dscp62map8021p: -> {7}
dscp63map8021p: -> {7}

```

2 Create the **packet-rule-record** to assign DSCP to CoS.

```

zSH> rule add dscptocos 1/1 1
Created packet-rule-record 1/1 (dscptocos)

```

3 Verify **packet-rule-record** 1/1.

```

zSH> get packet-rule-record 1/1
packet-rule-record 1/1
packetRuleType: ---> {dscptocos}
packetRuleValue: --> {1} <----- references dscp-to-cos profile 1
packetRuleValue2: -> {}
packetRuleValue3: -> {}
packetRuleValue4: -> {}
packetRuleValue5: -> {}
packetRuleValue6: -> {}
packetRuleValue7: -> {}

```

4 If necessary, the **dscp-to-cos** profile can be modified with the **update dscp-to-cos 1** command.

```

zSH> update dscp-to-cos 1
dscp-to-cos 1
Please provide the following: [q]uit.
dscp00map8021p: -> {0}:
dscp01map8021p: -> {0}:
dscp02map8021p: -> {0}:
dscp03map8021p: -> {0}:
dscp04map8021p: -> {0}:
dscp05map8021p: -> {0}:
dscp06map8021p: -> {0}:
dscp07map8021p: -> {0}:
dscp08map8021p: -> {1}:
dscp09map8021p: -> {1}:
dscp10map8021p: -> {1}:
dscp11map8021p: -> {1}:
dscp12map8021p: -> {1}:
dscp13map8021p: -> {1}:
dscp14map8021p: -> {1}:
dscp15map8021p: -> {1}:
dscp16map8021p: -> {2}:

```

```

dscp17map8021p: -> {2}:
dscp18map8021p: -> {2}:
dscp19map8021p: -> {2}:
dscp20map8021p: -> {2}:
dscp21map8021p: -> {2}:
dscp22map8021p: -> {2}:
dscp23map8021p: -> {2}:
dscp24map8021p: -> {3}:
dscp25map8021p: -> {3}:
dscp26map8021p: -> {3}:
dscp27map8021p: -> {3}:
dscp28map8021p: -> {3}:
dscp29map8021p: -> {3}:
dscp30map8021p: -> {3}:
dscp31map8021p: -> {3}:
dscp32map8021p: -> {4}:
dscp33map8021p: -> {4}:
dscp34map8021p: -> {4}:
dscp35map8021p: -> {4}:
dscp36map8021p: -> {4}:
dscp37map8021p: -> {4}:
dscp38map8021p: -> {4}:
dscp39map8021p: -> {4}:
dscp40map8021p: -> {5}:
dscp41map8021p: -> {5}:
dscp42map8021p: -> {5}:
dscp43map8021p: -> {5}:
dscp44map8021p: -> {5}:
dscp45map8021p: -> {5}:
dscp46map8021p: -> {5}:
dscp47map8021p: -> {5}:
dscp48map8021p: -> {6}:
dscp49map8021p: -> {6}:
dscp50map8021p: -> {6}:
dscp51map8021p: -> {6}:
dscp52map8021p: -> {6}:
dscp53map8021p: -> {6}:
dscp54map8021p: -> {6}:
dscp55map8021p: -> {6}:
dscp56map8021p: -> {7}:
dscp57map8021p: -> {7}:
dscp58map8021p: -> {7}:
dscp59map8021p: -> {7}:
dscp60map8021p: -> {7}:
dscp61map8021p: -> {7}:
dscp62map8021p: -> {7}:
dscp63map8021p: -> {7}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Deleting the packet-rule-record

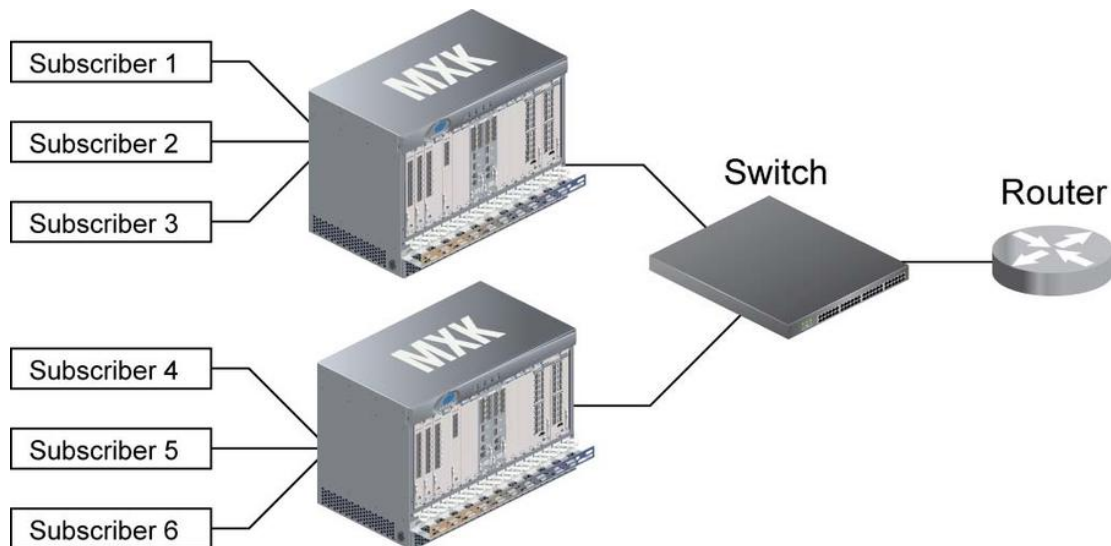
Delete the **packet-rule-record**.

```
zSH> delete packet-rule-record 1/1
packet-rule-record 1/1
1 entry found.
Delete packet-rule-record 1/1? [y]es, [n]o, [q]uit : yes
packet-rule-record 1/1 deleted.
```

Destination MAC swapping

The destination MAC swapping feature provides a security enhancement which prevents port-to-port communications between users sharing a VLAN for Internet access when the user-to-user traffic spans multiple MXK shelves as shown in [Destination MAC swapping on page 352](#).

Figure 48: Destination MAC swapping



When enabled, this feature modifies the destination MAC address portion of unicast frames (Ethernet frames not using a multicast or broadcast destination MAC) that traverse the MXK so that the destination MAC is changed to the MAC address of the next-hop router in the access network. This address modification ensures that all frames in the access network are forwarded to the access router regardless of how the frame originated. Broadcast, multicast, and Ethernet frames with a destination MAC address of the next hop router are forwarded without MAC swapping.

The MXK retrieves the MAC address of the next hop router to correctly swap into unicast frames through dynamically snooping DHCP ACK messages or a static user-specified entry.

- Dynamically snooping DHCP ACK messages

The MXK snoops DHCP ACK messages received on the bridge interface that is configured as the default (VLAN or default bridge). The source MAC address from this frame is swapped into for frames received on interfaces configured for destination MAC swapping. This address is stored in the database and persists across reboots. When a new DHCP ACK message is received in the same VLAN, its source is checked, and if different, the newer MAC address is used.

This option requires that DHCP server services are used in the network and that the next hop router is the default router between the MXK and the DHCP server.

- Static user-specified entry

The MXK inserts the user-specified valid 6-byte hexadecimal MAC address into unicast frames not matching the static entry.



Note: Destination MAC swapping is only supported on the uplink cards on the MXK.

Configuring destination MAC swapping

Use the **rule add** command to create either the dynamic or static destination MAC swapping rule:

```
rule add <dstmacswapdynamic|dstmacswapstatic> <groupindex/Memberindex> <MAC address>
```

The rule for dynamic MAC swapping does not have a parameter. The rule for static MAC swapping requires a parameter, the MAC address to match.

```
rule add dstmacswapdynamic groupindex/Memberindex
```

```
rule add dstmacswapstatic groupindex/Memberindex macaddress
```

dstmacswapdynamic or dstmacswapstatic

MAC addresses of the net hop router used to correctly swap into unicast frames through either dynamically snooping DHCP ACK messages or a static user-specifies entry.

Syntax `dstmacswapdynamic or dstmacswapstatic`

Options `dstmacswapdynamic`

Dynamic MAC swapping reads the destination MAC address from the default VLAN on the uplink to swap into the packet, so you just need to define which uplink bridge interface to associate with the rule.

dstmacswapstatic

Static MAC swapping requires a MAC address to be swapped into the packet which you must supply.

Example 1 For dynamic MAC swapping:

```
zSH> rule add dstmacswapdynamic 1/1
Created packet-rule-record 1/1 (dstmacswapdynamic)
```

Example 2 For static MAC swapping:

```
zSH> rule add dstmacswapstatic 2/1 08:00:20:bc:8b:8c
Created packet-rule-record 2/1 (dstmacswapstatic)
```

Example 3 View the rules.

```
zSH> rule show
```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30
		auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30
		auto-enable-interval (def) 300 600 1200
1/1	dstmacswapdynamic	00:00:00:00:00:00
2/1	dstmacswapstatic	08:00:20:bc:8b:8c

4 record(s) found

Bridge storm protection

This section describes the packet rule for bridge storm protection:

- [Bridge storm protection overview, page 355](#)
- [Default packet rule filters \(bridgestormdetect\), page 355](#)
- [Case 1: bridgestormdetect packet rule for discard, page 358](#)
- [Case 2: bridgestormdetect packet rule for discard + alarm, page 359](#)
- [Case 3: bridgestormdetect packet rule for discard + alarm + block, page 360](#)
- [Modify the default bridgestormdetect rules, page 361](#)
- [View detected packets statistics, page 363](#)
- [Unblock a bridge, page 366](#)

Bridge storm protection overview

The **bridgestormdetect** filter provides a way to analyze packets by capturing discarded packets when a certain threshold is reached and is configured only on the ingress of a bridge interface.

This packet rule will capture the first N packets after the target packets-per-second threshold is reached. Since all discarded packets are not captured, and there may be multiple interfaces with a bridge storm, some packets on the first interface with a bridge storm are captured, then some packets on the next interface with a bridge storm are captured, and so on.

The **rule add bridgestormdetect** command syntax is:

```
rule add bridgestormdetect <group/member> <discard | discardandalarm  
|discardandalarmandblock> <packets-per-second> [<consecutive-seconds>]
```

If the **rule add bridgestormdetect** command is configured with *discard*, only the *packets-per-seconds* is set.

If the **rule add bridgestormdetect** command is configured with *discardandalarm* or *discardandalarmandblock*, both the *packets-per-seconds* and the *consecutive-seconds* fields must be set.

If the card reboots, the captured packets are lost.

Default packet rule filters (bridgestormdetect)

Currently, default packet rules are created only for the **bridgestormdetect** filter.

The default **bridgestormdetect** rule is configured for *discard+alarm+block* with defined auto-enable intervals.

Rules for default packet rule *bridgestormdetect*

The rules for the default **bridgestormdetect** packet rule filters are:

- A default packet rule filter for **bridgestormdetect** is automatically defined and applied to *downlink*, *tls*, and *wire* bridge interfaces when a **bridgestormdetect** packet rule is not currently applied.
- If an eligible bridge type is configured with packet rules other than **bridgestormdetect**, the default **bridgestormdetect** rule is applied.
- The default packet rules are configured in group *0*.
- The group/member *0/1* **bridgestormdetect** rule is automatically applied to *downlink* bridge interfaces and rule *0/2* is automatically applied to *tls* and *wire* bridge interfaces.
- The default **bridgestormdetect** rule is not applied to other bridge types.

The default rules are always displayed with the **rule show** command:

```
zSH> rule show
      Group/Member                                Type Value(s)
-----
      Default dwn (0/1)                          bridgestormdetect discard+alarm+block pps 30 cs 30
                                                    auto-enable-interval (def) 300 600 1200
      Default tls/wire (0/2)                      bridgestormdetect discard+alarm+block pps 100 cs 30
                                                    auto-enable-interval (def) 300 600 1200
2 record(s) found
```

The **rule showuser default** command displays bridges with the default packet rule **bridgestormdetect**.

```
zSH> rule showuser default
      Group/Member                                Type      IfIndex  IfAddr
-----
      Default dwn (0/1)                          bridgestormdetect 1359 1-4-1-303-gponport-100/bridge
(ingress)
      Default dwn (0/1)                          bridgestormdetect 1362 1-4-1-501-gponport/bridge (ingress)
2 record(s) found
```

Disable the *bridgestormdetect* packet rules

The default **bridgestormdetect** rules can be disabled by entering the *disdefpktrules* keyword to the **options** parameter in **system 0**. Both default packet rules are disabled.

The default rules *0/1* and *0/2* cannot be deleted with the **rule delete** command.

```
zSH> rule delete 0/1
Not allowed to delete from default group index 0
```

Disabling the default *bridgestormdetect* packet rules

Update the **system 0** file.

```
zSH> update system 0
```



```

system 0
Please provide the following: [q]uit.
syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {false}:
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: ---> {false}:
configsyncpriority: ---> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: ---> {}:
configsyncstatus: -----> {syncinitializing}:
configsyncuser: -----> {}:
configsyncpasswd: -----> {** private **}: ** read-only **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {0.0.0.0}:
alternateipaddress: ---> {0.0.0.0}:
countryregion: -----> {us}:
primaryclocksource: ---> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {NONE(0)}: disdefpktrules <-----
reservedVlanIdStart: --> {0}:
reservedVlanIdCount: --> {0}:
snmpVersion: -----> {snmpv2}
persistentLogging: ----> {disabled}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Re-enabling the default bridgestormdetect packet rule

Update **system 0** by entering the *none 0* keyword to the **options** parameter.

```

zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:

```

```

setserialno: -----> {0}:
zmsexists: -----> {false}:
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: ---> {false}:
configsyncpriority: ---> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: ---> {}:
configsyncstatus: -----> {syncinitializing}:
configsyncuser: -----> {}:
configsyncpasswd: -----> {** private **}: ** read-only **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {0.0.0.0}:
alternateipaddress: ---> {0.0.0.0}:
countryregion: -----> {us}:
primaryclocksource: ---> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {disdefpktrules}: none 0 <-----
reservedVlanIdStart: --> {0}:
reservedVlanIdCount: --> {0}:
snmpVersion: -----> {snmpv2}
persistentLogging: ----> {disabled}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Case 1: bridgestormdetect packet rule for discard

Configuring a bridge discard

Configuring the *bridgestormdetect* packet rule for *discard*, means that when the packets exceed the packets-per-second threshold, the overall traffic on the bridge will be limited.

- 1 Enter the **rule add** command to create the *bridgestormdetect* packet rule for *discard* and set the packets-per-second threshold.

```

zSH> rule add bridgestormdetect 1/1 discard pps 20
Created packet-rule-record 1/1 (bridgestormdetect)

```

Verify the rule.

```

zSH> rule show
      Group/Member                                Type Value(s)

```

```

-----
Default dwn (0/1)          bridgestormdetect discard+alarm+block pps 30 cs 30
                             auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)     bridgestormdetect discard+alarm+block pps 100 cs 30
                             auto-enable-interval (def) 300 600 1200
                             1/1          bridgestormdetect discard pps 20
3 record(s) found

```

2 Apply the rule to a bridge interface.

```

zSH> bridge add 1-6-1-0/eth downlink vlan 100 tagged ipktrule 1
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-100/bridge

```

Verify the bridge.

```

zSH> bridge show
Orig
Type   VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
dwn          Tagged 100    1/6/1/0/eth    1-6-1-0-eth-100/bridge    UP  D 00:01:47:31:dc:1a
1 Bridge Interfaces displayed

```

Verify the rule 1/1 is applied to the bridge.

```

zSH> rule showuser
Group/Member      Type      IfIndex  IfAddr
-----
1/1              bridgestormdetect  1354    1-6-1-0-eth-100/bridge (ingress)
1 record(s) found

```

Case 2: bridgestormdetect packet rule for discard + alarm

Configuring a rule for discard + alarm

Configuring the *bridgestormdetect* packet rule for *discard + alarm*, means that when the packets exceeds the packets-per-second threshold over a configured number of seconds, the overall traffic on the bridge will be limited and a bridge storm alarm will be sent. When the bridge storm is cleared, a clearing alarm is sent.

- 1 Enter the **rule add** command to create the *bridgestormdetect* packet rule for *discard + alarm*.

```

zSH> rule add bridgestormdetect 2/1 discardandalarm pps 20 cs 10
Created packet-rule-record 2/1 (bridgestormdetect)

```

Verify the rule.

```

zSH> rule show
Group/Member      Type  Value(s)
-----
Default dwn (0/1)  bridgestormdetect discard+alarm+block pps 30 cs 30
                             auto-enable-interval (def) 300 600 1200

```

```
Default tls/wire (0/2)          bridgestormdetect discard+alarm+block pps 100 cs 30
                                auto-enable-interval (def) 300 600 1200
                                1/1          bridgestormdetect discard pps 20
                                2/1          bridgestormdetect discard+alarm pps 20 cs 10
4 record(s) found
```

2 Apply the rule to a bridge interface.

```
zSH> bridge add 1-6-2-0/eth downlink vlan 400 tagged ipktrule 2
Adding bridge on 1-6-2-0/eth
Created bridge-interface-record 1-6-2-0-eth-400/bridge
```

Verify the bridge.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
dwn   Tagged 100  1/6/1/0/eth  1-6-1-0-eth-100/bridge  UP  D 00:01:47:31:dc:1a
dwn   Tagged 400  1/6/2/0/eth  1-6-2-0-eth-400/bridge  UP
2 Bridge Interfaces displayed
```

Verify the rule 2/1 is applied to the bridge.

```
zSH> rule showuser
Group/Member  Type  IfIndex  IfAddr
-----
1/1          bridgestormdetect  1354  1-6-1-0-eth-100/bridge (ingress)
2/1          bridgestormdetect  1356  1-6-2-0-eth-400/bridge (ingress)
2 record(s) found
```

Case 3: bridgestormdetect packet rule for discard + alarm + block

Configuring the *bridgestormdetect* packet rule for *discard + alarm + block*, means that when the packets exceeds the packets-per-second threshold over a configured number of seconds, the overall traffic on the bridge will be completely blocked and a bridge storm alarm will be sent. When the bridge storm is cleared, a clearing alarm is sent.

The *bridgestormdetect* packet rule for *discard + alarm + block* automatically creates an **auto-enable-interval** parameter configured for 300 seconds, 600 seconds, and 1200 seconds. The first value indicates that the bridge will automatically unblock after 300 seconds (five minutes). The second value indicates that when the next bridge storm occurs, the bridge will unblock after 600 seconds (ten minutes), and after the third bridge storm detection, the bridge will unblock after 1200 seconds (20 minutes). After the third time, if the storm continues, the bridge remains blocked and must be unblocked through the CLI. See [Unblock a bridge, page 366](#).

Configuring a rule for discard + alarm + block

- 1 Enter the **rule add** command to create the *bridgestormdetect* packet rule for *discard + alarm + block*.

```
zSH> rule add bridgestormdetect 3/1 discardandalarmandblock pps 20 cs 10
Created packet-rule-record 3/1 (bridgestormdetect)
```

Verify the rule.

```
zSH> rule show
```

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 30 cs 30 auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30 auto-enable-interval (def) 300 600 1200
1/1	bridgestormdetect	discard pps 20
2/1	bridgestormdetect	discard+alarm pps 20 cs 10
3/1	bridgestormdetect	discard+alarm+block pps 20 cs 10 auto-enable-interval (def) 300 600 1200

5 record(s) found

2 Apply the rule to a bridge interface.

```
zSH> bridge add 1-6-3-0/eth downlink vlan 500 tagged ipktrule 3
Adding bridge on 1-6-3-0/eth
Created bridge-interface-record 1-6-3-0-eth-500/bridge
```

Verify the bridge.

```
zSH> bridge show
```

Type	Orig	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn	Tagged	100	1/6-1-0/eth	1-6-1-0-eth-100/bridge	UP	D 00:01:47:31:dc:1a
dwn	Tagged	400	1/6-2-0/eth	1-6-2-0-eth-400/bridge	DWN	
dwn	Tagged	500	1/6-3-0/eth	1-6-3-0-eth-500/bridge	DWN	

3 Bridge Interfaces displayed

Verify the rule 3/1 is applied to the bridge.

```
zSH> rule showuser
```

Group/Member	Type	IfIndex	IfAddr
1/1	bridgestormdetect	1354	1-6-1-0-eth-100/bridge (ingress)
2/1	bridgestormdetect	1356	1-6-2-0-eth-400/bridge (ingress)
3/1	bridgestormdetect	1357	1-6-3-0-eth-500/bridge (ingress)

3 record(s) found

Modify the default bridgestormdetect rules

The default parameters in the **bridgestormdetect** rule can be modified by the user.

The syntax for the **rule modify bridgestormdetect** is:

```
rule modify bridgestormdetect <group/member>
[<discard | discardandalarm | discardandalarmandblock >]
[pps <packets-per-second>] [cs <consecutive-seconds>]
[auto-enable-interval <param0> [<param1> [<param2>]]]
```

The **rule modify** command allows you to disable or change the *auto-enable-interval* values as well as the threshold *pps* and *cs*.

Modify default bridgestormdetect pps and cs values

The **bridgestormdetect** *discardandalarmandblock* packet rule blocks the bridge interface when packets exceed a level configured by the *pps* over time set by the *cs* value.

The default values for *pps* and *cs* in default 0/1 and 0/2 differ due to higher normal traffic on *tls* and *wire* bridges.

The range for consecutive alarm seconds values is 5 to 30 seconds.

Modifying default pps and cs values

- 1 Enter the **rule modify bridgestormdetect** command to change the default values.

```
zSH> rule modify bridgestormdetect 0/1 discardandalarmandblock pps 25 cs 25
Modified packet-rule-record 0/1 (bridgestormdetect)
```

- 2 Verify the changes.

```
zSH> rule show
Group/Member                                     Type Value(s)
-----
Default dwn (0/1)                               bridgestormdetect discard+alarm+block pps 25 cs 25
auto-enable-interval (def) 300 600 1200
Default tls/wire (0/2)                           bridgestormdetect discard+alarm+block pps 100 cs 30
auto-enable-interval (def) 300 600 1200
2 record(s) found
```

Default bridgestormdetect auto-enable-interval values

The default *auto-disable-interval* parameter sets the time in seconds when the bridge is unblocked and allowed to pass traffic at 300, 600, and 1200 seconds.

When a bridge interface is blocked the first time, it is unblocked after 300 seconds. The second time, if the storm continues, the interface is unblocked after 600 seconds. The third time, if the storm continues, the bridge interface is unblocked at 1200 seconds. After the third time, if the storm continues, the bridge remains blocked and must be unblocked through the CLI. See [Unblock a bridge, page 366](#).

The *auto-enable-interval* times in seconds can be modified or disabled.

Modifying the auto-enable-interval values

- 1 Enter the **rule modify bridgestormdetect** command to change the default values for *auto-enable-interval*.

```
zSH> rule modify bridgestormdetect 0/1 discardandalarmandblock pps 25 cs 25
auto-enable-interval 60 300 600
```

Modified packet-rule-record 0/1 (bridgestormdetect)

2 Verify the changes.

zSH> **rule show**

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 25 cs 25 auto-enable-interval 60 300 600
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30 auto-enable-interval (def) 300 600 1200

2 record(s) found

Disabling the default auto-enable-interval

Entering the value *0* to the first field of the **auto-enable-interval** parameter disables the re-enable traffic feature of **bridgestormdetect**.

- 1 Enter the **rule modify bridgestormdetect** command to disable the **auto-enable-interval**.

zSH> **rule modify bridgestormdetect 0/2 discardandalarmandblock pps 100 cs 30 auto-enable-interval 0**

Modified packet-rule-record 0/2 (bridgestormdetect)

The bridge interface will be blocked and must be unblocked through CLI.
See [Unblock a bridge on page 366](#)

- 2 Verify the change.

zSH> **rule show**

Group/Member	Type	Value(s)
Default dwn (0/1)	bridgestormdetect	discard+alarm+block pps 25 cs 25 auto-enable-interval 60 300 600
Default tls/wire (0/2)	bridgestormdetect	discard+alarm+block pps 100 cs 30 auto-enable-interval 0

2 record(s) found

View detected packets statistics

Viewing detected packets statistics

The **bridge stats interface/type** command sorts and displays the detected packets into unicast, multicast, or broadcast and displays the number of alarms sent.

zSH> **bridge stats 1-6-1-0-eth-100/bridge**

Interface Name	Received Packets			Transmitted Packets				Storm Detect Packets			
	UCast	MCast	BCast	UCast	MCast	Bcast	Error	UCast	MCast	Bcast	Alarm
1-6-1-0-eth-100/bridge	--	--	--	--	--	--	--	0	0	0	0

1 Bridge Interfaces displayed

View the packets

Use the **bridge capture show** command to view which interfaces had a bridge storm and how many packets were captured.

The Packet column shows the number of packets captured, and the Count column displays the number of packets allowed to be captured.

Each interface having a bridge storm will capture fewer packets. The first interface that has a bridge storm can capture eight packets, the next interface that has a bridge storm can capture six packets, and so on.

Viewing the packets

You must connect to the line card before using the **bridge capture show** command.

- 1 Connect to the line card by entering **connect** and the slot number of the line card.

```
zSH> connect 3
Connecting to shelf: 1, slot: 3 .....
Connection established.
```

- 2 Enter the **bridge capture show** command to view which interfaces had a bridge storm and how many packets were captured.

```
zSH> bridge capture show
```

Interface Name	Packet	Count
bond-0502-efmbond	8/	8
<Queue Empty>	0/	6
<Queue Empty>	0/	4
<Queue Empty>	0/	2

- 3 Enter the **bridge capture dump interface/type** command to view the captured packets.

```
zSH> bridge capture dump bond-0502-efmbond/bridge bond-0502-efmbond, IfIndex =
46979 # tick = 0x0000001f2275ef54
00000000: ff ff ff ff ff ff 00 00 00 00 00 01 08 00 45 00 ".....E."
00000010: 00 2e 96 0d 00 00 40 11 d9 b0 0a 01 01 01 ff ff ".....@....."
00000020: ff ff 04 00 04 01 00 1a 00 00 00 04 20 00 00 04 "....."
00000030: 00 00 96 0d 88 ff 66 a5 77 00 99 5a db db db db ".....f.w..Z..."
00000040: 05 c1 46 60 00 00 00 51 00 fe c0 94 00 00 00 38 "..F^...Q.....8"
00000050: ed ed ed ed ed ed ed ed 05 be bc 28 05 bf 9d 58 ".....(...X"
00000060: ed ed ed ed ed ed ed ed ed ed ed ed ed ed ed "....."
00000070: ed ed ed ed ed ed ed ed ed ed ed ed ed ed ed "....."
# bond-0502-efmbond, IfIndex = 46979
# tick = 0x0000001f2275f8f3
00000000: ff ff ff ff ff ff 00 00 00 00 00 01 08 00 45 00 ".....E."
00000010: 00 2e 96 10 00 00 40 11 d9 ad 0a 01 01 01 ff ff ".....@....."
00000020: ff ff 04 00 04 01 00 1a 00 00 00 04 20 00 00 04 "....."
00000030: 00 00 96 10 88 ff 70 f5 77 00 8f 0a db db db db ".....p.w....."
00000040: 05 bf 6e 40 00 00 00 51 00 fe c0 94 00 00 00 28 "..n@...Q.....("
```



```

00000050: ed ed ed ed ed ed ed ed 05 bf 73 a8 05 c1 09 68 ".....s....h"
00000060: ed ed ed ed ed ed ed ed ed ed ed ed ed ed ed "....."
00000070: ed ed ed ed ed ed ed ed ed ed ed ed ed ed ed "....."
# bond-0502-efmbond, IfIndex = 46979
# tick = 0x00000001f2276015f
00000000: ff ff ff ff ff ff 00 00 00 00 00 01 08 00 45 00 ".....E."
00000010: 00 2e 96 13 00 00 40 11 d9 aa 0a 01 01 01 ff ff ".....@....."
00000020: ff ff 04 00 04 01 00 1a 00 00 00 04 20 00 00 04 "....."
00000030: 00 00 96 13 88 ff 7b 45 77 00 84 ba db db db db ".....{Ew....."
00000040: 05 bf 72 a0 00 00 00 50 00 fe c0 94 00 00 00 24 "r....P.....$"
00000050: 00 00 00 01 00 00 00 00 00 00 00 06 00 00 00 4f ".....O"
00000060: ed ed ed ed ed ed ed ed 00 00 00 7c ed ed ed ed ".....|...."
00000070: 00 00 00 00 db db db db db db db db db db db "....."
# bond-0502-efmbond, IfIndex = 46979
# tick = 0x00000001f227641d4
00000000: ff ff ff ff ff ff 00 00 00 00 00 01 08 00 45 00 ".....E."
00000010: 00 2e 96 15 00 00 40 11 d9 a8 0a 01 01 01 ff ff ".....@....."
00000020: ff ff 04 00 04 01 00 1a 00 00 00 04 20 00 00 04 "....."
00000030: 00 00 96 15 88 ff 82 25 77 00 7d da db db db db ".....%w.}....."
00000040: 05 c2 06 20 00 00 00 51 00 fe c0 94 00 00 00 38 "...
...Q.....8"
00000050: ed ed ed ed ed ed ed ed 05 c0 6c 48 05 c0 0f e8 ".....lH...."
00000060: ed ed ed ed ed ed ed ed ed ed ed ed ed ed ed "....."
00000070: ed ed ed ed ed ed ed ed ed ed ed ed ed ed ed "....."
# bond-0502-efmbond, IfIndex = 46979
# tick = 0x00000001f2277c395
00000000: ff ff ff ff ff ff 00 00 00 00 00 01 08 00 45 00 ".....E."
00000010: 00 2e 96 18 00 00 40 11 d9 a5 0a 01 01 01 ff ff ".....@....."
00000020: ff ff 04 00 04 01 00 1a 00 00 00 04 20 00 00 04 "....."
00000030: 00 00 96 18 88 ff 8c 75 77 00 73 8a db db db db ".....uw.s...."
00000040: 05 bf 6f d0 00 00 00 51 00 fe c0 94 00 00 00 38 "o....Q.....8"
00000050: ed ed ed ed ed ed ed ed 05 be f7 d8 05 bf 30 18 ".....0."
00000060: ed ed ed ed ed ed ed ed ed ed ed ed ed ed ed "....."
00000070: ed ed ed ed ed ed ed ed ed ed ed ed ed ed ed "....."
# bond-0502-efmbond, IfIndex = 46979
# tick = 0x00000001f22793e41
00000000: ff ff ff ff ff ff 00 00 00 00 00 01 08 00 45 00 ".....E."
00000010: 00 2e 96 1b 00 00 40 11 d9 a2 0a 01 01 01 ff ff ".....@....."
00000020: ff ff 04 00 04 01 00 1a 00 00 00 04 20 00 00 04 "....."
00000030: 00 00 96 1b 88 ff 96 c4 77 00 69 3b db db db db ".....w.i;...."
00000040: 05 bf 9d 90 00 00 00 51 00 fe c0 94 00 00 00 38 ".....Q.....8"
00000050: ed ed ed ed ed ed ed ed 05 c1 23 d8 05 bf 9e 98 ".....#....."
00000060: ed ed ed ed ed ed ed ed ed ed ed ed ed ed ed "....."
00000070: ed ed ed ed ed ed ed ed ed ed ed ed ed ed ed "....."
# bond-0502-efmbond, IfIndex = 46979
# tick = 0x00000001f25008cf3
00000000: ff ff ff ff ff ff 00 00 00 00 00 01 08 00 45 00 ".....E."
00000010: 00 2e 96 20 00 00 40 11 d9 9d 0a 01 01 01 ff ff "...
..@....."
00000020: ff ff 04 00 04 01 00 1a 00 00 00 04 20 00 00 04 "....."
00000030: 00 00 96 20 88 ff a7 f4 77 00 58 0b db db db db "...
...w.X....."
00000040: 05 bf 2f b0 00 00 00 51 00 fe c0 94 00 00 00 28 " ../....Q.....("
00000050: ed ed ed ed ed ed ed ed 05 bf 70 38 05 c0 2e f8 ".....p8...."

```

```

00000060: ed ed ed ed ed ed ed ed ed ed ed ed ed ed ed "....."
00000070: ed ed ed ed ed ed ed ed ed ed ed ed ed ed ed "....."
# bond-0502-efmbond, IfIndex = 46979
# tick = 0x0000001f250209bf
00000000: ff ff ff ff ff ff 00 00 00 00 00 01 08 00 45 00 ".....E."
00000010: 00 2e 96 23 00 00 40 11 d9 9a 0a 01 01 01 ff ff "...#..@....."
00000020: ff ff 04 00 04 01 00 1a 00 00 00 04 20 00 00 04 "....."
00000030: 00 00 96 23 88 ff b2 44 77 00 4d bb db db db db "...#...Dw.M...."
00000040: 05 bf d6 e0 00 00 00 50 00 fe c0 94 00 00 00 28 ".....P..... ("
00000050: 05 c8 0f e0 c5 0b 4b 0c 00 00 00 00 c5 0b 4b 0c ".....K.....K."
00000060: 00 00 b7 83 05 c1 e7 50 05 bf 30 60 05 cc a7 a0 ".....P..0`...."
00000070: 00 00 00 00 05 be 6b 20 db db db db db db db db ".....k ....."

```



Note: For customers who want to view output in a packet capture tool such as Wireshark, copy the output into a notepad file, then run the *text2pcap* application. The output should then be in a viewable state.

- 4 Enter the **bridge capture clear -all** command to clear all the interfaces with bridge storms, then verify the output with the **bridge capture show** command.

You can also enter the bridge capture clear *interface/type* command to clear individual bridge interfaces.

```
zSH> bridge capture clear -all
```

```
1/3-admin@Mxk-50.2.31> bridge capture show
```

Interface Name	Packet Count
<Queue Empty>	0/ 8
<Queue Empty>	0/ 6
<Queue Empty>	0/ 4
<Queue Empty>	0/ 2

- 5 Close the connection to the line card by entering the **exit** command.

```
zSH> exit
```

Connection closed.

Unblock a bridge

Unblocking a bridge

Use the **bridge unblock interface/type** command to unblock a blocked bridge interface configured with the **bridgestormdetect** packet rule *discard + alarm + block*.

Enter the **bridge unblock** command.

```
zSH> bridge unblock 1-6-1-0-eth-100/bridge
```

Access Control List (ACL)

This section describes the Access Control List (ACL) packet rules and includes:

- [ACL packet rule filtering rules on the MXK, page 367](#)
- [ACL packet rule filtering variables, page 367](#)
- [ACL filtering options, page 368](#)
- [Configure ACL packet rules, page 370](#)

ACL packet rule filtering rules on the MXK

The ACL filters allow you to deny or allow packets based on packet characteristics. The ACL filters are configured using packet rules.

The following rules apply to ACL filtering on the MXK:

- ACL packet rules work only on the ingress port of a line card and do not block traffic on the egress port (to the subscriber).
- ACL packet rules work on *downlink* and *tls* bridge types by configuring the bridge with the keyword *ipktrule*. For example,

```
bridge add interface/type downlink | tls vlanid
ipktrule
```
- ACL packet rules only work on packets sent to the CPU.
- ACL packet rules can only be used to prevent or allow MAC address learning and are useful when configuring service authorization.

ACL packet rule filtering variables

The ACL filtering options also include the ability to allow or deny packets on the ingress port of line cards.

ACL configuring options are:

- Ethernet types ARP, IP, VLAN, PPPoE discovery or PPPoE data, or as defined by hex or numeric bits. See [ethtype on page 368](#).
- destination MAC address, either broadcast address or as defined by address bits in hex. See [dstmac \(destination MAC address\) and bcast on page 368](#).
- source MAC address, either broadcast address or as defined by address bits in hex. See [srcmac \(source MAC address\) and bcast on page 368](#).
- SLAN
- VLAN
- IP protocols: ICMP, IGMP, TCP, UDP
- source IP port: source IP address in IP packets

- destination IP port: telnet, DHCP server, DHCP client
- allow all or deny all packets

ACL filtering options

This section describes the ACL filtering variables:

- [allow or deny based on source and destination MAC addresses, page 368](#)
- [allow or deny based on Ethernet types, page 368](#)
- [allow or deny based on source IP/port, page 370](#)

allow or deny based on source and destination MAC addresses

all (allow and deny). **allow all** is used in combination with specific **deny** list rules to create a list of packets not allowed. **deny all** is used in combination with specific **allow** list rules to create a list of packets allowed.

dstmac (destination MAC address) and bcast. Use *dstmac* rule to allow or deny packets to pass based on the destination MAC address.

There are a maximum of five destination MAC address filters per interface and up to 1000 destination MAC address filters per system.

The *bcast* variable is the broadcast address.

hh:hh:hh:hh:hh:hh[/Bits] (addr bytes in hex)

srcmac (source MAC address) and bcast. Use *srcmac* rule to allow or deny packets to pass based on the source MAC address of the packet.

There are a maximum of five source MAC address filters per interface and up to 1000 source MAC address filters per system.

The *bcast* variable is the broadcast address.

hh:hh:hh:hh:hh:hh[/Bits] (addr bytes in hex)

slan (outer VLAN ID). Matches outer VLAN ID (slan)

vlan (inner VLAN ID). Matches inner VLAN ID (vlan).

allow or deny based on Ethernet types

ethtype . Use the *ethtype* rules to allow or deny packets using numeric codes with the *ethtype* rules. The 13th and 14th octets of an Ethernet (IEEE 802.3) packet after the preamble consists of the Ethernet type or the IEEE 802.3 length field.

More common Ethernet types, such as IP or ARP, may be designated by name.

Preamble	Destination MAC addr	Source MAC addr	Ether Type	Payload	CRC32	Interframe gap
7 octets	6 octets	6 octets	2 octets	46-1500 octets	4 octets	12 octets

Numeric values must be hexadecimal. Prepend the “0x” prefix to the Ethernet type numeric code. For example, the IP Ethernet Type code 0800 would be 0x0800.



Note: Access Control List has several IPv6 options for rule add and rule deny:

- ipv6 (v6 version of IP address)
- icmp6 (IP proto 58)
- srcipv6 (v6 version of srcip)
- dstipv6 (v6 version of dstip)
- dhcp6s (DHCPv6 server port 547)
- dhcp6c (DHCPv6 client port 546)

Using the numeric keyword for an *eth*type allows you to filter based on any Ethernet type as shown in [Table 27](#).

Table 27: Numeric codes for common Ethernet types

Ethernet Type	Keyword	Numeric code
ARP (Address Resolution Protocol)	arp	0x0806
IP	ip	0x0800
VLAN	vlan	0x8100
PPPoE discovery	pppoedisc	0x8863
PPPoE data	pppoedata	0x8864
	0xhhhh[/Bits] or nnnnn[/Bits]	



Note: PPPoE filtering only, not PPPoA filtering is supported.

allow or deny based on source IP/port

ipproto. The *ipproto* filtering rules match the IP and UDP protocols in IP packets. [Table 28](#) describe the protocol identifiers.

Table 28: IP and UDP protocols

Supported IP and UDP protocols	protocol
icmp	01
igmp	02
tcp	06
udp	17

srcip . Matches the source IP address in IP packets.

dstip . Matches the destination IP address in IP packets.

srcport. Matches the source IP port in IP packets.

dstport. Matches the destination IP port in IP packets.

Table 29: IP ports in IP packets

Type	Port
telnet	Telnet port 23
dhcps	DHCP server port 67
dhcpc	DHCP client port 68

Configure ACL packet rules

This section describes ACL packet rule behavior and how to create the ACL packet rules:

- [Create allow or deny packet rules, page 370](#)
- [The order of multiple ACL filters on an interface, page 371](#)
- [ACL statistics and clear statistics commands, page 374](#)

Create allow or deny packet rules

When creating a rule that denies a source MAC address, an additional rule must be created to define the behavior of the first rule. For example, when a rule is created to deny access to a source MAC address, an *allow* rule must also be created to allow all other MAC addresses to pass.

For example,

```
zSH> rule add deny 1/1 srcmac 00:01:02:03:04:05
Created packet-rule-record 1/1 (deny)
```

Because the addition of this first rule would not only deny access to packets with that particular source MAC address but all packets, an *allow* rule must also be created. In this way access to packets with that particular source MAC address is denied and access to all other packets is allowed. you would need to add another rule to allow all packets.

The allow rule *must* exist in the same group and the deny rule.

For example

```
zSH> rule add deny 1/1 srcmac 00:01:02:03:04:05
Created packet-rule-record 1/1 (deny)
```

```
zSH> rule add allow 1/2 all
Created packet-rule-record 1/2 (allow)
```

In most (if not all) applications of the ACL rules, the **allow all** or **deny all** will be the last rule in the group. If an **allow all** or **deny all** rule is not present, an implicit **deny all** rule is executed.

Please note that the **allow all** and **deny all** rules will not affect the regular transmission of broadcast and multicast frames on *downlink* bridge interfaces, so normal bridge functions will continue. Since *tls* bridge interfaces normally allow all packets, the **allow all** and **deny all** rules will affect all the packets.

The order of multiple ACL filters on an interface

While each filter works independently of other filters and may be applied to the same interface the filter are supposed to work together for maximum flexibility. When multiple filters are applied to an interface, rule order is important.

Rule order is defined in the membership index. Rules with the lowest **memberIndex** have the highest priority. Execution of the filtering terminates upon the first successful match.

For example, when packet rules are created in this order in a member index,

```
zSH> rule add deny 1/10 srcmac 06:05:04:03:02:01
Created packet-rule-record 1/10 (deny)
```

```
zSH> rule add allow 1/30 all
Created packet-rule-record 1/30 (allow)
```

and a packet is encountered which has a source MAC address of 06:05:04:03:02:01 and a destination MAC address of 00:01:02:03:04:05, the packet will be blocked (discarded) because the *deny* rule was matched. If the order were different, so that the allow rule had a **groupIndex/memberIndex** of 1/10 then the packet would be allowed.

If **allow all** was 1/10, all of the packets would be allowed and none of the other rules would ever be executed, so the careful ordering of the ACL rules is important.

It is good practice to leave available spots for the ordering of the ACL packet rules, so that rules can be added before or between existing rules without needing to change the numbers of existing rules.

Deny rules based on wild cards within the MAC address. You can create a rule to filter in or out packets based on portions of the MAC address. The most common filter would work like the **bridgeforbidoui** rule. While ACLs may behave like the **bridgeforbidoui** rule, they provide a powerful mechanism for filtering with wild cards.

Creating a rule which works like the **bridgeforbidoui** rule but with wild cards, which significant bits to filter for a MAC address are defined. The **bridgeforbidoui** rule denies access based on the Organizationally Unique Identifier (OUI). An organization's OUI is the first bytes of the MAC address.

For example, creating the rule,

```
zSH> rule add deny 1/1 srcmac 00:01:02:00:00:00/24
Created packet-rule-record 1/1 (deny)
```

denies access for packets from a device whose source MAC address starts with 00:01:02. It is these first three bytes (24 bits) which supply the forbid OUI for the device.



Note: The **bridgeforbidoui** rule will not change and is being kept for legacy reasons, so if you have **bridgeforbidoui** rules, you need not change them.

If you need to deny access based on the first four bytes, you would create a rule such as,

```
zSH> rule add deny 1/1 srcmac 00:01:02:03:00:00/16
Created packet-rule-record 1/1 (deny)
```

Even though the examples show 00s for the bits for which we do not care about their value, the /24 defines the filter bits. The examples use 00 for the bits whose value is not cared about as a programming practice.

When no mask is wanted, use the /48 on the MAC address, or leave the mask off.

Deny all multicast IP traffic. Multicast traffic has its own OUI, 01:00:5e, making it easy to deny multicast IP traffic.

```
zSH> rule add deny 1/1 dstmac 01:00:5e:00:00:00/24
Created packet-rule-record 1/1 (deny)
```




Note: Downlink bridge interfaces drop upstream multicast traffic by default.

Limit traffic to PPPoE. zSH> `rule add allow 1/10 ethtype pppoe-disc`

Created packet-rule-record 1/10 (allow)

zSH> `rule add allow 1/20 ethtype pppoe-data`

Created packet-rule-record 1/20 (allow)

zSH> `rule add deny 1/30 all`

Created packet-rule-record 1/30 (deny)

Note that the **deny all** is not necessary, but is a best programming practice.

Create rules with AND operations. When rules are combined in a single command, the rules are ANDed, so to limit traffic to PPPoE discovery broadcast and data packets for a specific MAC address you put them in a single command:

zSH> `rule add allow 1/20 dstmac 00:01:02:03:04:05 ethtype pppoe-disc`

Created packet-rule-record 1/20 (allow)

zSH> `rule add allow 1/30 dstmac 00:01:02:03:04:05 ethtype pppoe-data`

Created packet-rule-record 1/30 (allow)

zSH> `rule add deny 1/100 all`

Created packet-rule-record 1/100 (deny)

Use Ethernet type codes. You may use the common name or numeric Ethernet type code.

To limit traffic to PPPoE packets and two destination MAC addresses:

zSH> `rule add allow 1/20 dstmac 00:01:02:03:04:05 ethtype pppoe-disc`

Created packet-rule-record 1/20 (allow)

zSH> `rule add allow 1/30 dstmac 00:01:02:03:04:05 ethtype pppoe-data`

Created packet-rule-record 1/30 (allow)

zSH> `rule add allow 1/40 ethtype 0x8863 dstmac 00:01:02:03:04:06`

Created packet-rule-record 1/40 (allow)

zSH> `rule add allow 1/50 dstmac 00:01:02:03:04:06 ethtype 0x8864`

Created packet-rule-record 1/50 (allow)

zSH> `rule add deny 1/100 all`

Created packet-rule-record 1/100 (deny)

Note that order of the commands in the single rule command is not important.

ACL statistics and clear statistics commands

ACL rule add commands. The **ruleType** for ACL commands is **allow** or **deny** (other than **bridgeforbidoui** which is an implied deny without explicitly stating as the other ACL commands).

```
rule add <ruleType> <groupIndex/memberIndex> <value
[value] ...>
```

The next parameter is one of the following keywords: **dstmac**, **srcmac**, **ethtype**, or **all**.

```
rule add <add|deny> <<srcmac macaddress> <dstmac
macaddress> <ethtype ethtype>|all>
```

Table 30: ACL ruleType keywords

Keyword	Value(s)	Bits (default)
dstmac	hh:hh:hh:hh:hh:hh broadcast (ff:ff:ff:ff:ff:ff)	<0..48> (48)
srcmac	hh:hh:hh:hh:hh:hh	<0..48> (48)
ethtype	numeric arp (0x0806) ip (0x0800) pppoediscovery (0x8863) pppoedata (0x8864)	<0..16> (16)
all	all packet conditions will be addressed by the final default condition (whether allow or deny).	

Please note that once a single ACL **allow** or **deny ruleType** is used, there is an implicit unstated **deny all** rule. You can block all traffic if you do not add an **allow all** rule at the end of the group.

ACL rule show command. Syntax:

```
rule show acl [<groupIndex>[/<memberIndex>]]
```

Omission of **groupIndex/memberIndex** displays all ACL rules. Omission of just **memberIndex** displays all ACL rules matching the given **groupIndex**.

Examples:

```
zSH> rule show acl
Group/Member      Type      HitCount Value(s)
-----
1/20              allow      0 dstmac 00:01:02:03:04:05
                  ethtype pppoedisc (0x8863)
1/30              allow      0 dstmac 00:01:02:03:04:05
                  ethtype pppoedata (0x8864)
```

```

          1/40          allow          0 dstmac 00:01:02:03:04:06
                                ethtype pppoedisc (0x8863)
          1/50          allow          0 dstmac 00:01:02:03:04:06
                                ethtype pppoedata (0x8864)
          1/100         deny           0 all
5 record(s) found

```

zSH> **rule show acl 1**

Group/Member	Type	HitCount	Value(s)
1/20	allow	0	dstmac 00:01:02:03:04:05 ethtype pppoedisc (0x8863)
1/30	allow	0	dstmac 00:01:02:03:04:05 ethtype pppoedata (0x8864)
1/40	allow	0	dstmac 00:01:02:03:04:06 ethtype pppoedisc (0x8863)
1/50	allow	0	dstmac 00:01:02:03:04:06 ethtype pppoedata (0x8864)
1/100	deny	0	all

5 record(s) found

zSH> **rule show acl 1/40**

Group/Member	Type	HitCount	Value(s)
1/40	allow	0	dstmac 00:01:02:03:04:06 ethtype pppoedisc (0x8863)

1 record(s) found

The **rule show acl** commands display only ACL related rules, i.e. those with rule types *allow*, *deny*, or *bridgeforbidoui*. The **rule show acl** commands display a HitCount column which shows the number of times a rule was matched. Counts are held in a 64 bit format. Both HOST and NP (or equivalent) generated counts are aggregated together. If count exceeds 1T (10**12), display will show "n.nnnT", if count exceeds 1G (10**9), display will show "n.nnnG", else it will display a 10 digit number.

zSH> **rule show acl**

Group/Member	Type	HitCount	Value(s)
1/1	allow	0	dstmac bcast (ff:ff:ff:ff:ff:ff) ethtype pppoedisc (0x8863)
1/2	allow	1234567890	dstmac 00:01:02:03:04:05 ethtype pppoedisc (0x8863)
1/10	deny	517691	all
19/2	bridgeforbidoui	1.001G	00:81:80
19/5	bridgeforbidoui	2.123T	00:80:80

The older existing rule **bridgeforbidoui** is technically a deny specific rule, so it is displayed with the ACL rules.

The **bridgeforbidoui** rule provides a means to block devices based on their OUI which are incompatible on the network or for other security reasons. The same filtering may be done with the allow/deny ACL rules, though you do not need to change existing rules. The **bridgeforbidoui** rule is kept for backward compatibility.

ACL rule stats. The **rule stats acl** command displays or clears the ACL stats.

Syntax:

```
rule stats acl [<groupIndex>[/<memberIndex>]]
```

Omission of groupIndex/memberIndex displays all ACL rules. Omission of just **memberIndex** displays all ACL rules matching the given **groupIndex**.

Running ACL statistics

After applying the ACL rule on the ingress of a *downlink* or *tls* bridge, you must connect to the slot of the line card, then run the **rule stats acl** command.

 **Note:** Before connecting to the line card, the user must have *debug* privileges. See [User account administration on page 68](#).

- 1 Connect to the line card by entering the connect command with the shelf and slot number.

```
zSH> connect 1 4
Connecting to shelf: 1, slot: 4 .....
Connection established.
1/4-zSH>
```

- 2 Enter the **rule stats acl** command on the line card.

```
1/4-zSH> rule stats acl
Group/Member      Type      HitCount Value(s)
-----
1/20              allow     0 dstmac 00:01:02:03:04:05
                  ethtype pppoedisc (0x8863)
1/30              allow     0 dstmac 00:01:02:03:04:05
                  ethtype pppoedata (0x8864)
1/40              allow     0 dstmac 00:01:02:03:04:06
                  ethtype pppoedisc (0x8863)
1/50              allow     0 dstmac 00:01:02:03:04:06
                  ethtype pppoedata (0x8864)
1/100             deny      0 all
5 record(s) found
```

The **rule stats acl** command can also be entered on the group number. Display is identical to that of **rule show acl** command.

```
1/4-zSH> rule stats acl 1
Group/Member      Type      HitCount Value(s)
-----
1/20              allow     0 dstmac 00:01:02:03:04:05
                  ethtype pppoedisc (0x8863)
1/30              allow     0 dstmac 00:01:02:03:04:05
                  ethtype pppoedata (0x8864)
1/40              allow     0 dstmac 00:01:02:03:04:06
                  ethtype pppoedisc (0x8863)
1/50              allow     0 dstmac 00:01:02:03:04:06
```

```

                    ethertype pppoeata (0x8864)
                    1/100          deny          0 all
5 record(s) found

```

The **rule stats acl** command can also be entered on the group and member number.

```

1/4-zSH> rule stats acl 1/40
Group/Member      Type      HitCount Value(s)
-----
                    1/40          allow          0 dstmac 00:01:02:03:04:06
                    ethertype pppoeata (0x8864)
1 record(s) found

```

3 Close the connection to the line card when finished.

```

1/4-zSH> exit
Connection closed.

```

Clearing ACL statistics

The **rule stats acl clear** command clears the hit counts on all selected ACL rules.

Syntax:

```
rule stats acl clear [<groupIndex>[/<memberIndex>]]
```

1 Connect to the line card by entering the connect command with the shelf and slot number

```

zSH> connect 1 4
Connecting to shelf: 1, slot: 4 .....
Connection established.

```

2 Enter the **rule stats acl clear** command(s).

Omission of the group and member index clears all ACL rules. Omission of just member index clears all ACL rules matching the given group index. Entering the group and member index clears the statistics for both the group and the member.

```

1/4-zSH> rule stats acl clear
5 record(s) cleared

1/4-zSH> rule stats acl clear 1
5 record(s) cleared

1/4-zSH> rule stats acl clear 1/40
1 record(s) cleared

```

3 Close the connection to the line card when finished.

```

1/4-zSH> exit
Connection closed.

```

Additional bridging services

This section describes:

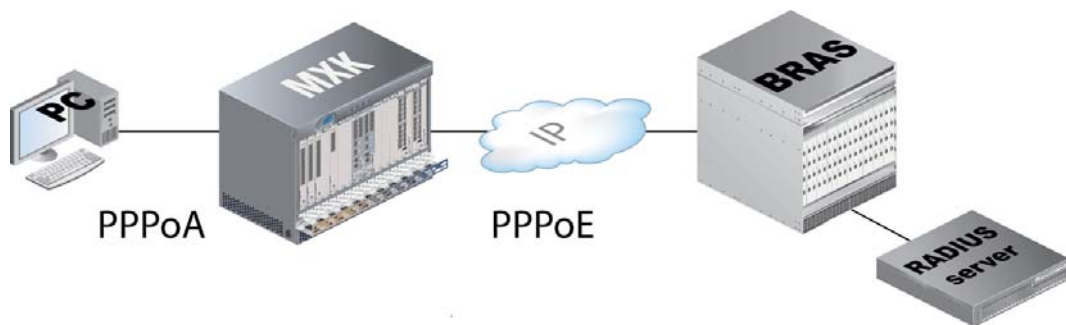
- [PPPoA - PPPoE interworking on bridges, page 378](#)
- [Rapid Spanning Tree Protocol \(RSTP\), page 381](#)
- [Multiple Spanning Tree Protocol \(MSTP\) on the MXK, page 389](#)
- [Shaping Traffic: Class of Service Queuing, page 406](#)
- [COS and SCOS replacement on Ethernet frames, page 409](#)
- [“Denial of Service” prevention, page 411](#)
- [Bridging differences between the MALC and MXK, page 412](#)

PPPoA - PPPoE interworking on bridges

The MXK supports PPPoA to PPPoE interworking for connections to a Broadband Remote Access Server (BRAS) using a PPP tunnel. Upon detecting PPPoA traffic, the MXK initiates a PPPoE session with the Broadband Remote Access Server (BRAS). PPP traffic between the CPE and the BRAS is tunneled over this PPPoE session. The MXK autosenses the type of PPPoA encapsulation as either VCMUX or LLC.

An inactivity timeout occurs when a lack of activity is detected on the PPPoA connection for 30-80 seconds, while upstream PPPoE packets are received. When this occurs, the PPPoE session is terminated.

Figure 49: PPPoA to PPPoE interworking



Enabling PPPoA to PPPoE interworking

PPPoA – PPPoE interworking is added by enabling PPPoA on an ADSL downlink bridge.

The **bridge add** command supports enabling PPPoA interworking from the CLI. This example creates a downlink bridge on the interface *interface/adsl* with VLAN 500 and enables the PPPoA to PPPoE feature.

```

zSH> bridge add 1-10-1-0/adsl vc 0/35 td 1 downlink vlan 500 pppoa
Adding bridge on 1-10-1-0/adsl
  
```

Created bridge-interface-record 1-10-1-0-adsl-0-35/bridge

This command automatically updates the **bridge-interface record** profile.



Note: The following message may appear if the CPE device is not properly configured for PPPoA connections.

```
FEB 01 15:59:22: error   : 1/1/8 : bridge:
_afsmChkRcvEncaps(): l=1811: tNetTask:
AFSM-6313: port 1-9-24-0-adsl-0-35 misconfigured
for PPPoA
```

Verifying PPPoA – PPPoE interworking

1 Verify the PPPoA parameter in the **bridge-interface-record**

```
zSH> get bridge-interface-record 1-10-1-0-adsl-0-35/bridge
bridge-interface-record 1-10-1-0-adsl-0-35/bridge
vpi: -----> {0}
vci: -----> {35}
vlanId: -----> {500}
stripAndInsert: -----> {true}
customARP: -----> {false}
filterBroadcast: -----> {false}
learnIp: -----> {false}
learnUnicast: -----> {false}
maxUnicast: -----> {0}
learnMulticast: -----> {false}
forwardToUnicast: -----> {false}
forwardToMulticast: -----> {false}
forwardToDefault: -----> {true}
bridgeIfCustomDHCP: -----> {false}
bridgeIfIngressPacketRuleGroupIndex: -> {0}
vlanIdCOS: -----> {0}
outgoingCOSOption: -----> {disable}
outgoingCOSValue: -----> {0}
s-tagTPID: -----> {0x8100}
s-tagId: -----> {0}
s-tagStripAndInsert: -----> {true}
s-tagOutgoingCOSOption: -----> {s-tagdisable}
s-tagIdCOS: -----> {0}
s-tagOutgoingCOSValue: -----> {0}
mcastControlList: -----> {}
maxVideoStreams: -----> {0}
isPPPoA: -----> {true}
floodUnknown: -----> {false}
floodMulticast: -----> {false}
bridgeIfEgressPacketRuleGroupIndex: --> {0}
bridgeIfTableBasedFilter: -----> {NONE(0)}
bridgeIfDhcpLearn: -----> {NONE(0)}
mvrVlan: -----> {0}
vlan-xlate-from: -----> {0}
```

```
slan-xlate-from: -----> {0}
bridge-type: -----> {pppoa}
```

2 Use the **bridge show** command to display the state of the PPPoA session.

When the PPPoA port status is UP, the BRAS MAC address and PPPoE session ID are also displayed.

PPPoA port states are:

- PENDING (PND)
The bridge port has not yet bound with the driver during initialization. This state is for all bridges. A bridge cannot transition back to this state.
- READY (RDY)
Waiting for PPPoA packet to initiate PPPoE discovery.
- UP
The PPPoA port is active. The BRAS MAC address and PPPoE session ID will also be displayed.
- DOWN (DWN)
The PPPoA port is down
- DISCVRY (DSC)
PPPoE discovery initiated. Waiting for session ID to be obtained.
PPPoA port is pending.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
poa          500    1/10/1/0/adsl    1-10-1-0-adsl-0-35/bridge  PND D 00:01:47:36:59:aa
1 Bridge Interfaces displayed
```

PPPoA port is ready.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
poa          500    1/10/1/0/adsl    1-10-1-0-adsl-0-35/bridge  RDY D 00:01:47:36:59:aa
1 Bridge Interfaces displayed
```

PPPoA port is up.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
poa          500    1/10/1/0/adsl    1-10-1-0-adsl-0-35/bridge  UP D 00:01:47:36:59:aa
1 Bridge Interfaces displayed
```


Rapid Spanning Tree Protocol (RSTP)

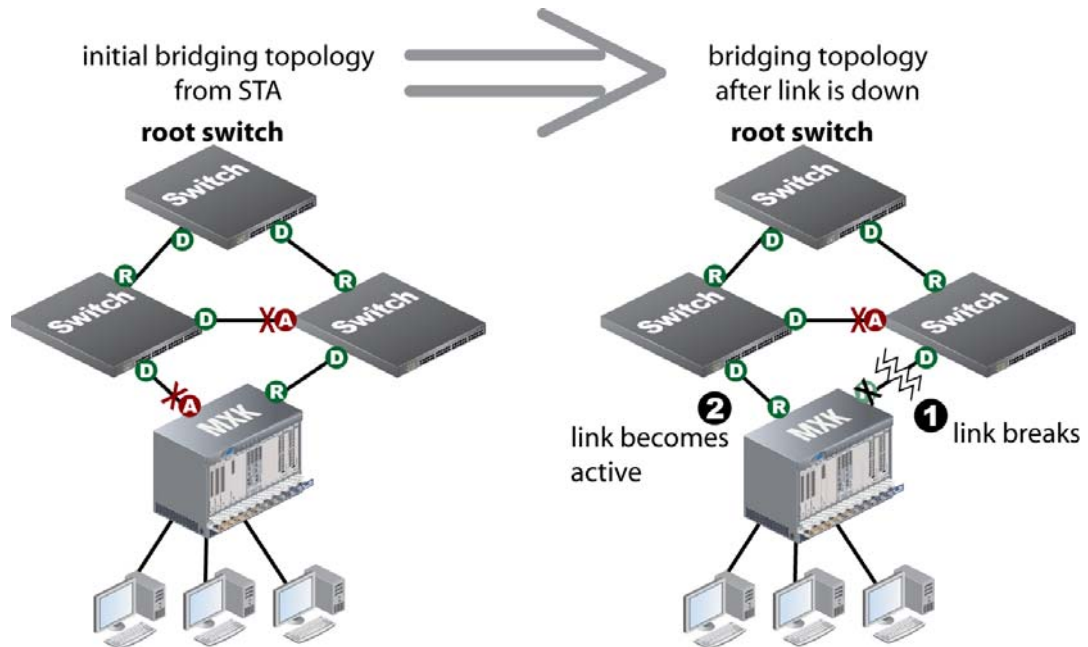
RSTP (802.1W) is an evolution of the Spanning Tree Protocol (STP, IEEE 802.1D). STP links network segments and eliminates one of the difficulties of configuring bridge topologies — bridge loops. There still can only be one active path. Once RSTP is configured for a bridged network, the Spanning Tree Algorithm (STA) analyzes the network and determines which links should be active or not. The STA defines the links by configuring the ports.

In the bridged network the root bridge is selected. The STA sends out messages — Bridge Protocol Data Units (BPDU) — to determine the least cost path to the root bridge. From this analysis the port roles are determined.



Note: RSTP is supported on simplex uplinks only (not redundant uplinks).

Figure 50: The STA defines the initial bridging topology and later adjusts



RSTP port role

There are five port roles assigned by the STA to the port:

- **ROOT:** Root port

The root port is the closest to the root switch (also as root bridge. The root bridge is the only switch/bridge in the network that does not have a root port because it is the central bridge and root ports are defined by their relationship to the root bridge). The root port will receive the best BPDU from the root switch on a bridge.

In [Figure 50](#), the root ports are designated with “R.”

For the STA to determine the root port for a device, five RSTP priority parameters are compared in the following priority sequence:

- 1) root bridge priority
- 2) root path cost
- 3) designated bridge priority
- 4) designated port ID
- 5) port priority

Only one RSTP port can be chosen as the root port per device. The port with the lowest value of RSTP priority parameters wins. If the first RSTP priority parameter have the same values on the ports, then the system will compare the next one, until it finds the root port.

- DSNT: Designated port

The designated port is the best port to send BPDU from the RSTP device to networked device.

In [Figure 50](#), the designated ports are designated with “D.”

- ALT: Alternate port

The alternate port is a port that is blocked because it is receiving more useful BPDUs from another bridge. The alternate port can change to an active root port.

In [Figure 50](#), the alternate ports are designated with “A” and are shown as blocked.

- BKP: Backup port

The backup port is a port that is blocked because it is receiving more useful BPDUs from the same bridge it is on. A backup port is only providing connectivity to the same network segment, so it cannot change to a root port.

- N/A: Not available

It means RSTP is not in the functional state yet. It usually will appear right after system bootup.

To view RSTP port roles, use **bridge show** command or **rstp-bridge show** command.

RSTP port state

IEEE 802.1w defines three port states in RSTP:

- DIS: RSTP discarding
- LRN: RSTP learning (a transitional state)
- FWD: RSTP forwarding (a normal operational state)

In operation there is no difference between a port with state DIS and one with state LRN as they both discard frames and do not learn MAC addresses. Ports which are blocking must keep transmitting BPDUs to retain maintain its port role and port state.

To show the RSTP port states, use **bridge show** command or **rstp-bridge show** command.

RSTP on uplinks

Rapid Spanning Tree Protocol (RSTP, IEEE 802.1W) is supported on upstream interface on the following MXK uplink cards:

- MXK-UPLINK-2X10G-8X1GE
- MXK-UPLINK-8X1GE
- MXK-UPLINK-4X1GE-CU
- MXK-UPLINK-4X1GE



Note: Interface 1-a-1-0/eth can not be used for RSTP. This interface is for inband management only.

Configuring RSTP on uplink bridges

The following example configures RSTP on uplink bridges.

- 1 Create RSTP uplink bridges on MXK upstream ports *1-a-4-0/eth* and *1-a-5-0/eth*:

Use **stp-bridge add interface/type uplink vlan x <tagged>** to add a VLAN interface to the upstream interface.

```
zSH> stp-bridge add 1-a-4-0/eth uplink vlan 500
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-500/bridge
Bridge-path added successfully
```

```
zSH> stp-bridge add 1-a-5-0/eth uplink vlan 500
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-500/bridge
Bridge-path added successfully
```

The bridge-path is automatically created with the parameter *default*.

Even if the parameter *tagged* is not specified, the uplink bridge is considered a tagged bridge and the bridge will appear as tagged when using **bridge show**.

- 2 Show the bridges, enter:

```
zSH> bridge show
      Orig
Type VLAN/SLAN  VLAN/SLAN      Physical      Bridge      St  Table Data
-----
```

```

upl          Tagged 500 1/a/4/0/eth          ethernet4-500/bridge      BLK
upl          Tagged 500 1/a/5/0/eth          ethernet5-500/bridge      FWD S VLAN 500 default STP: ROOT
2 Bridge Interfaces displayed

```

Port *1-a-5-0* has been chosen as the root port, which is an active uplink port is receiving and forwarding packets. Port *1-a-4-0* is the alternate port, which is blocked and discarding packets.

3 To get detail RSTP information, use **stp-bridge show** command.

```

zSH> stp-bridge show
Bridge is running IEEE 802.1W RSTP
Bridge ID has priority 36000, address 00:01:47:14:c3:00
Configured: hello=2, forward=15, max_age=20
This bridge is the ROOT of the topology
1 bridge(s) present first-> ethernet4-500:
    Port is DOWN!
1 bridge(s) present first-> ethernet5-500:
    is a DESIGNATED PORT in FORWARDING state
Root bridge has priority 36000, address 00:01:47:14:c3:00
Designated bridge has priority 36000, address 00:01:47:14:c3:00
Designated Port id is 144:144, root path cost is 0
Timers: forward delay is 15, hello time is 2, message age is 0
sync: 0 synced: 1 reRoot: 0 rrWhile: 0 operEdge: 0 fdWhile: 0
learn: 1 forward: 1 agreed: 0 learning: 1 forwarding: 1 updtInfo: 0 selected: 1

```

Five RSTP priority parameters in these two ports will be compared in this sequence: Root bridge priority -> Root path cost -> Designated bridge priority -> Designated port ID -> Port priority.

In the above example, the value of the root bridge priority parameter is same on the two ports. Then, system compares the root path cost, since ethernet5-500 has the lower root path cost value 0, it becomes the root port.

4 If the first four RSTP priority parameters are the same, then the system compares the last parameter- port priority. The port with the lowest port priority wins. The port priority will be displayed when use **get stp-bind <profile-storage-key>** command, and can be changed use **update stp-bind <profile-storage-key>** command.

To verify the port priority in the **stp-bind** profile, enter:

```

zSH> get stp-bind ethernet4
stp-bind ethernet4/linegroup/0
portPriority: -> {128}

zSH> get stp-bind ethernet5
stp-bind ethernet5/linegroup/0
portPriority: -> {144}

```

To change the port priority in the **stp-bind** profile, enter:

```

zSH> update stp-bind ethernet4
stp-bind ethernet4/linegroup/0

```

```
Please provide the following: [q]uit.
portPriority: -> {128}: 160
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

- 5 To show the global RSTP parameters in the **stp-params** profile, use **get stp-params <profile-storage-key>** command.

```
zSH> get stp-params 0
stp-params 0
name: -----> {}
revision: -----> {0}
bridgePriority: -> {36000}
forceVersion: ---> {2}
fwdDelay: -----> {15}
helloTime: -----> {2}
migrateTime: ----> {3}
txHoldCount: ----> {3}
maxAge: -----> {20}
```

- 6 Delete the **stp-bridge(s)** on the ports.

```
zSH> stp-bridge delete ethernet4-500/bridge
Bridge-path deleted successfully
ethernet4-500/bridge delete complete

zSH> stp-bridge delete ethernet5-500/bridge
Bridge-path deleted successfully
ethernet5-500/bridge delete complete
```

RSTP rlinks

With the RSTP rlink in a ring configuration, instead of having a second redundant cloud link at each device, traffic can proceed through the other SLMS devices in the same network, which has its own uplink bridge.

See [Figure 51](#) for an RSTP rlink ring topology. In this example, there is the mixed use of MALC and MXK in a network. Each MALC and MXK has a bridge interface with the characteristics of an uplink bridge enabled on the port, and an intralink bridge on another port. With RSTP rlink enabled on the intralink bridge, the intralink interface designated B2 on the MXK will be blocked, preventing looped bridge traffic. Traffic from the root switch arriving on MXK A1 would be checked for destination MAC match for local ports (downlinks) and if a match is not found, the packet would be dropped. Traffic from downstream bridges on MXK would be sent upstream towards the root switch out the interface B1. Traffic from downstream bridges on MALC would be sent upstream towards the root switch out the interface A1

Figure 51: RSTP rlink ring topology

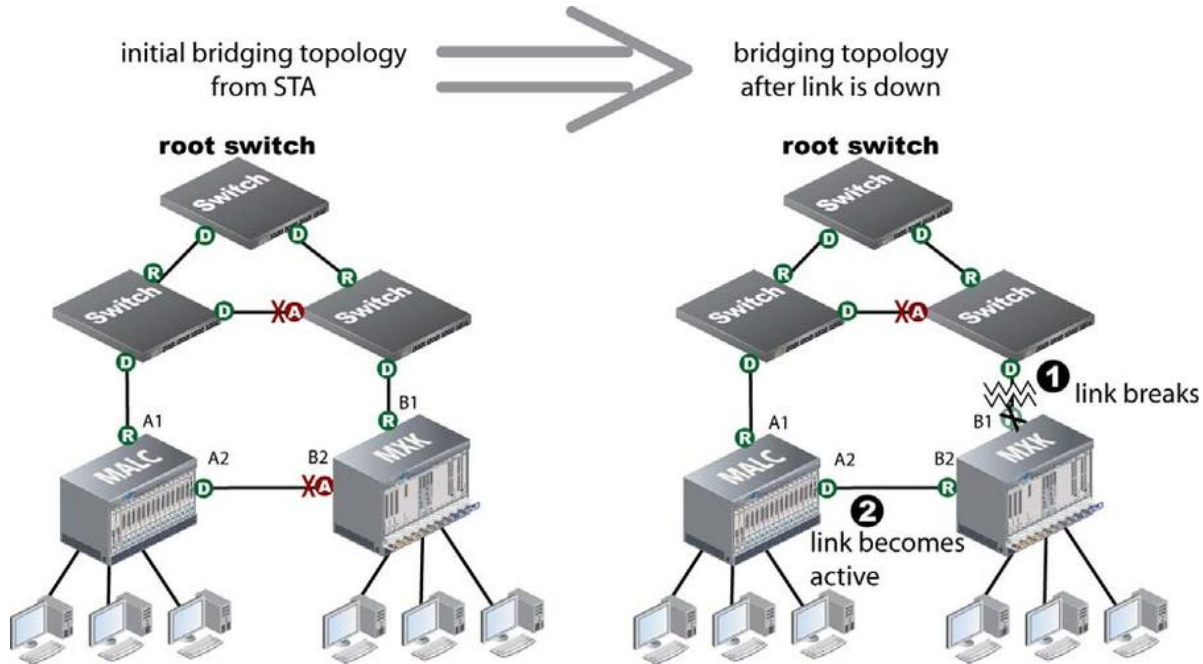
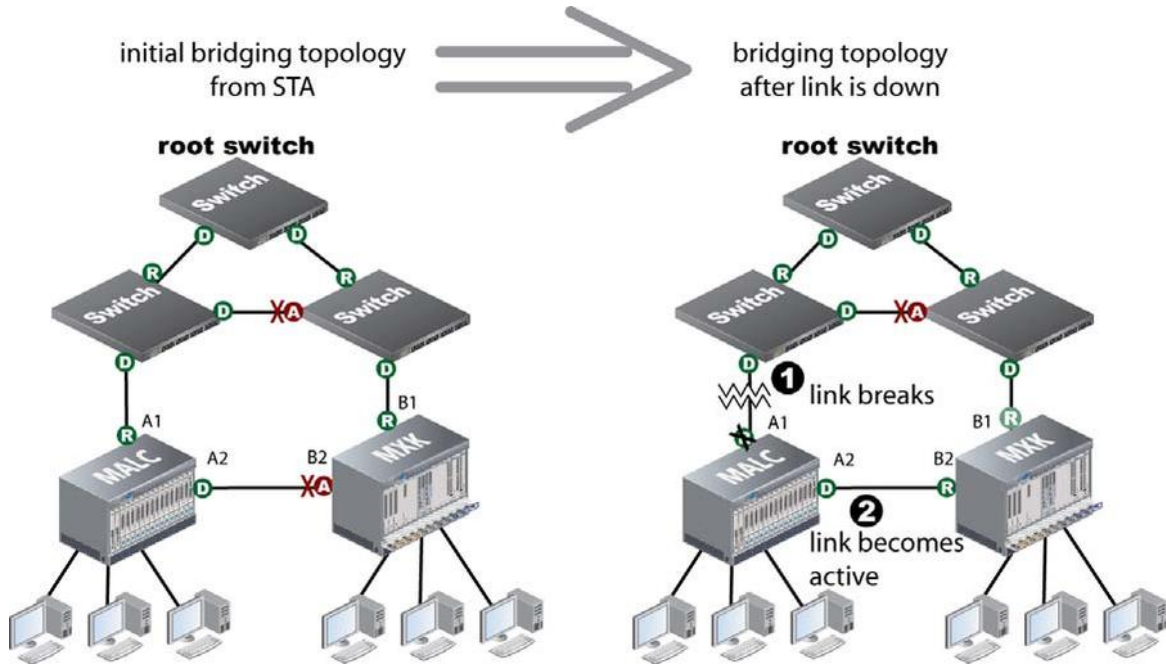


Figure 51 also shows that if the connection from MXK to the root switch becomes unavailable, then the RSTP ring protocol will take the port B2 on the MXK out of the blocking state and into a forwarding state. Traffic from downlink bridges on MXK will no longer leave on B1. Instead, downstream traffic will be forwarded on B2 heading towards A2, and then sent upstream towards the root switch out the MAC's root port interface A1.

Figure 52: RSTP rlink with a different downed link



Configuring RSTP rlinks

The configuration procedures for the RSTP rlink topologies are listed below.



Note: That this example show RSTP rlinks configured on both uplink and intralink ports on the MALC and MXK. You can also configure pure RSTP on the uplink port, and configure RSTP rlink on the intralink port.

- 1 As shown in [Figure 51](#), on the MALC, to configure RSTP rlinks on uplink and intralink bridges, perform the following tasks:
 - a Create RSTP rlink on upstream port A1 (1-1-2-0) and intralink port A2 (1-1-3-0) with **stp-bridge add interface/type rlink vlan id <tagged>**.

```
zSH> stp-bridge add 1-1-2-0/eth rlink vlan 500
Adding bridge on 1-1-2-0/eth
Created bridge-interface-record ethernet1-2-500/bridge
```

```
zSH> stp-bridge add 1-1-3-0/eth rlink vlan 500
Adding bridge on 1-1-3-0/eth
Created bridge-interface-record ethernet1-3-500/bridge
```

If the parameter *vlan id* is not specified, VLAN 0 is used. And if parameter *tagged* is not specified, the uplink bridge is considered a tagged bridge.

- b Create the bridge-paths for the rlink bridges using **bridge-path add interface/type global-rlink**.

```
zSH> bridge-path add ethernet1-3-500/bridge global-rlink
Bridge-path added successfully
Bridge-path added successfully
```

```
zSH> bridge-path add ethernet1-2-500/bridge global-rlink
Bridge-path added successfully
Bridge-path added successfully
```

- c View the bridge-paths.

```
zSH> bridge-path show
```

VLAN/SLAN	Bridge	Address
Global ethernet1-3-500/bridge		Default
Global ethernet1-3-500/bridge		Intralink
Global ethernet1-2-500/bridge		Default
Global ethernet1-2-500/bridge		Intralink

- d Show the baseline of the system, enter:

```
zSH> bridge show
```

Type	VLAN	Bridge	St	Table	Data
rlk	Tagged 500	ethernet2-500/bridge	FWD	S	Global default STP: ROOT
rlk	Tagged 500	ethernet3-500/bridge	DIS	STP:	ALT

Port A1 (1-1-2-0) has been chosen as the root port, which is an active uplink port in the forwarding state. Port A2 (1-1-3-0) is the intralink port and blocked by RSTP rlink topology to prevent loop. The state for this port is discarding. The role for this port is alternate.

- 2 On the MXK, to configure RSTP rlinks on uplink and intralink bridges, perform the following tasks:
 - a To create RSTP rlink on upstream port B1(1-a-4-0) and intralink port B2 (1-a-5-0):

```
zSH> stp-bridge add 1-a-4-0/eth rlink vlan 500
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-500/bridge
Bridge-path added successfully
Bridge-path added successfully

zSH> stp-bridge add 1-a-5-0/eth rlink vlan 500
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-500/bridge
Bridge-path added successfully
Bridge-path added successfully
```

- b Verify the bridge paths created, enter:

```
zSH> bridge-path show
VLAN/SLAN  Bridge                                     Address
-----
500 ethernet4-500/bridge                    Intralink
500 ethernet4-500/bridge                    Default, Age: 3600, MCAST Age: 150, IGMP Query
Interval: 70, Flap Mode: Default
500 ethernet5-500/bridge                    Intralink
500 ethernet5-500/bridge                    Default, Age: 3600, MCAST Age: 150, IGMP Query
Interval: 70, Flap Mode: Default
```

- c Show the baseline of the system, enter.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
rlk          Tagged 500  1/a/4/0/eth  ethernet4-500/bridge  BLK
rlk          Tagged 500  1/a/5/0/eth  ethernet5-500/bridge  FWD S VLAN 500 Intralink STP: ROOT
2 Bridge Interfaces displayed
```

Port B1 (1-a-5-0) has been chosen as the root port, which now is the closest port towards the root switch in terms of the root path cost. It can receive the best BPDUs from the root switch. Port B2 (1-a-4-0) is the intralink port has the designated port role, it can send and forward the best BPDUs.

- 3 As shown in [Figure 52](#), if the connection between the MALC uplink port A1 to the root switch is broken, the intralink port A2 on the MALC will be blocked and start to forward traffic from downlink bridges to MXK intralink port B2, since the MXK is the closest device to the root switch now.

- a** On the MALC, verify uplink port A1(1-1-5-0) is down, intralink port A2 (1-1-4-0) is in the forwarding state and takes over the role of root port, enter.

```
zSH> bridge show
Type VLAN      Bridge                               St  Table Data
-----
rlk Tagged 500  ethernet5-500/bridge                DWN
rlk Tagged 500  ethernet4-500/bridge                FWD S Global default STP: ROOT
```

- b** On the MXK, the port states and port roles will be same as before.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical      Bridge                               St  Table Data
-----
rlk             Tagged 500  1/a/4/0/eth   ethernet4-500/bridge                BLK
rlk             Tagged 500  1/a/5/0/eth   ethernet5-500/bridge                FWD S VLAN 500 Intralink STP: ROOT
2 Bridge Interfaces displayed
```

- 4** If you want to delete an RSTP rlink bridge, make sure to delete the uplink bridge path on bridge first, then delete the stp-bridge on the port.

- a** To delete the bridge path on MALC, use **bridge-path delete interface/bridge global-rlink** command.

```
zSH> bridge-path delete ethernet2-500/bridge rlink
Delete complete
Delete complete
```

To delete the bridge on MALC, use **stp-bridge delete interface/bridge** command.

```
zSH> stp-bridge delete ethernet2-500/bridge
ethernet2-500/bridge Delete complete
```

- b** To delete the bridge on MXK, use **stp-bridge delete interface/bridge** command.

```
zSH> stp-bridge delete ethernet4-500/bridge
Bridge-path deleted successfully
ethernet4-500/bridge delete complete
```

```
zSH> stp-bridge delete ethernet5-500/bridge
Bridge-path deleted successfully
ethernet5-500/bridge delete complete
```

Multiple Spanning Tree Protocol (MSTP) on the MXK

This section covers the implementation of MSTP on the MXK:

- [MSTP overview, page 390](#)
- [MSTP instances, page 390](#)
- [MSTP port role, page 390](#)

- [MSTP port states, page 391](#)
- [MSTP network routers, page 393](#)
- [MSTP network topology planning, page 393](#)
- [MSTP network topology components, page 393](#)
- [MSTP ring configuration, page 395](#)
- [MSTP ring operation, page 402](#)
- [MSTP ring IP on a bridge in-band device management, page 405](#)

MSTP overview

Multiple Spanning Tree Protocol (MSTP) on the MXK includes both IEEE 802.1S Multiple Spanning Tree Protocol (MSTP) and IEEE 802.1w Rapid Spanning Tree Protocol (RSTP). MSTP allows the grouping of VLANs to be mapped to multiple spanning tree instances (forwarding paths)

RSTP (Rapid Spanning Tree Protocol) on the MXK is configured per interface even when multiple VLANs are configured on the interface. This means that if four VLANs are configured on an interface on a port which is the active root port, and a loop is detected on just one of the VLANs, the entire port is blocked and all the data is switched to the alternate port which changes from a blocked state to become the active root port.

MSTP on the MXK differs from RSTP in that MSTP is configured on the VLAN and not on the interface. Therefore, when a fault is detected on an instance, only that VLAN is put into a blocked state and traffic is forwarded to a forwarding path.

MSTP allows multiple forwarding paths for data traffic. Traffic can leave the switch in either direction in the ring.

MSTP instances

Multiple Spanning Tree Instance(s) (MSTI) support groups of VLANs. Each MSTI can be configured with different root switches and different STP parameters.

MSTP port role

There are five port roles assigned by the STA to the port:

- **ROOT:** Root port

The root port is determined by the switch to be the most efficient way to pass traffic in the MSTP ring.

To determine the root port for a device, five MSTP priority parameters are compared in the following priority sequence:

- 1) root bridge priority

- 2) root path cost
- 3) designated bridge priority
- 4) designated port ID
- 5) port priority

Only one MSTP port can be chosen as the root port per device. The port with the lowest value of MSTP priority parameters wins. If the first MSTP priority parameter have the same values on the ports, then the system will compare the next one, until it finds the root port.

- DSNT: Designated port

A designated port is a port that has a lower priority than its root port.

- ALT: Alternate port

The alternate port is a backup port.

- BKP: Backup port

The backup port is a port that is blocked because it is receiving more useful BPDUs from the same bridge it is on. A backup port is only providing connectivity to the same network segment, so it cannot change to a root port.

- N/A: Not applicable

It means RSTP is not in the functional state yet. It usually will appear right after system bootup.

- Master

Not supported on Zhone devices.

To view MSTP port roles, use **bridge show** command.

MSTP port states

IEEE 802.1w defines three port states in MSTP:

- LRN: MSTP learning (a transitional state when the stp-bridge is first configured). For example,

```
zSH> bridge show
      Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
rlk          Tagged 100  1/1/2/0/eth      1-1-2-0-eth-100/bridge  LRN  STP: ROOT
1 Bridge Interfaces displayed
```

- FWD: MSTP forwarding (a normal operational state). For example.

```
zSH> bridge show
      Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
rlk          Tagged 100  1/1/2/0/eth      1-1-2-0-eth-100/bridge  FWD S VLAN 100 default STP: ROOT
```

1 Bridge Interfaces displayed

- DIS: MSTP discarding and traffic is not forwarding to the next device in the ring. For example,

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
rlk          Tagged 100  1/1/2/0/eth      1-1-2-0-eth-100/bridge  FWD S VLAN 100 default STP: ROOT
rlk          Tagged 100  1/1/3/0/eth      1-1-3-0-eth-100/bridge  DIS  STP: ALT
2 Bridge Interfaces displayed
```

In operation there is no difference between a port with state DIS and one with state LRN as they both discard frames and do not learn MAC addresses. Ports which are blocking must keep transmitting BPDUs to maintain its port role and port state.

To show the MSTP port states, use **bridge show** command:

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
rlk          Tagged 100  1/1/2/0/eth      1-1-2-0-eth-100/bridge  FWD S VLAN 100 default STP: ROOT
rlk          Tagged 100  1/1/3/0/eth      1-1-3-0-eth-100/bridge  DIS  STP: ALT
2 Bridge Interfaces displayed
```

or **stp-bridge show** command:

```
zSH> stp-bridge show
Bridge is running IEEE 802.1S (MSTP)
-- TreeID 0 --- (numTrees=3)
Bridge ID has priority 36864, address 00:01:47:d9:99:a0
  lostCistRoot=0 lostMstiRoot=0 alt2Root[0,0]
Configured: hello=2, forward=15, max_age=20 hops=20
Root port is 0, externalCost=20002 internalCost=20000
  1 bridge(s) present:
tree=0(0xea76dd8) is a ROOT PORT in FORWARDING state prtState[]= 0xea76e44
Root bridge has priority 24577, address f8:66:f2:0d:3c:41
Designated bridge has priority 32768, address 2c:36:f8:b3:c2:80
Designated Portid is 32788, externalCost=20002 internalCost=0
  1 bridge(s) present:
tree=0(0xea77e00) is a ALTERNATE PORT in DISCARDING state prtState[]= 0xea77e6c
Root bridge has priority 24577, address f8:66:f2:0d:3c:41
Designated bridge has priority 36864, address 00:01:47:22:99:f8
Designated Portid is 128, externalCost=20002 internalCost=40000
```

If a VLAN on the forwarding port goes down, the system switches to the alternate port which then becomes ROOT and forwards the packets to the node. For example, when Port 2 with VLAN 100 goes down, Port 3 with VLAN 100 becomes the forwarding port.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
```

```
-----  
rlk          Tagged 100 1/1/2/0/eth          1-1-2-0-eth-100/bridge      ADN  
rlk          Tagged 100 1/1/3/0/eth          1-1-3-0-eth-100/bridge      FWD S VLAN 100 default STP: ROOT  
2 Bridge Interfaces displayed
```

MSTP network routers

The routers upstream from the MXK must be configured to accept the data coming from the MSTP bridges in your network configuration.

MSTP network topology planning

When implementing MSTP on the MXK and any other devices in the MSTP ring, you *must* carefully design the network topology *before* creating MSTP bridges with the **stp-bridge add** command.

MSTP network topology components

The components of the MSPT network topology are:

- [The stp-params 0 profile, page 393](#)
- [The mstp-instance profile, page 394](#)
- [The stp-bind profile, page 395](#)
- [The STP bridges, page 395](#)

The stp-params 0 profile

The **stp-params 0** profile defines the MSTP region, the bridge priority, and the force version. There is just one **stp-params 0** profile configuration for each device in the network topology.

The **stp-params 0** file for each of the devices in the MSTP network *must* have the same MSTP region, bridge priority, and force version. This is because each **stp-bridge add** command will reference the same parameter definitions in the **stp-params 0** file.

A typical **stp-params 0** file for MSTP:

```
zSH> get stp-params 0  
stp-params 0  
name: -----> {Region1}  
revision: -----> {0}  
bridgePriority: -> {36864}  
forceVersion: ---> {3}  
fwdDelay: -----> {15}  
helloTime: -----> {2}  
migrateTime: ----> {3}  
txHoldCount: ----> {3}  
maxAge: -----> {20}
```

[Table 31](#) defines the parameters in the **stp-params 0** profile.

The user configurable parameters in the **stp-params 0** profile are *name*, *bridgePriority*, and *forceVersion*.

Table 31: stp-params 0 profile parameters

Parameter	Description
name	Field must be set to use MSTP, use the name of the bridge as a key.
revision	This parameter is used if you are running MSTP only. The MXK does not currently support any revisions to MSTP, so revision 0 is default. Default: 0
bridgePriority	The priority ID that will be advertised for this bridge. Must be a multiple of 4096. Default: 36864
forceVersion	The protocol to initiate with. 3- MSTP 2- RSTP 0- STP
fwdDelay	The delay used by STP bridges to transition Root and Designated ports to Forwarding. Default: 15
helloTime	The interval between periodic transmissions of Configuration Messages by designated ports. We only support a hello time of 2 currently. Default: 2
migrateTime	The initial value of the mdelayWhile and edgeDelayWhile timers. 3 is the only supported value for this timer. Default: 3
txHoldCount	The transmit hold count is used by the Port Transmit state machine to limit transmission rate. Default: 3
maxAge	The maximum age of the information transmitted by the bridge when it is the Root Bridge. Default: 20

The mstp-instance profile

The **mstp-instance** profile binds the instance and the VLAN ID.

An MXK can support up to fifty instances.

When planning the MSTP network, the **mstp-instance** for every VLAN *must* match on each device in the network. This is because a key is generated based on the region name and the **mstp-instance**. If a device does not have an mstp-instance, then the key that is generated will not match the key on the other devices.

This is because when a link in the MSTP network goes down, that state becomes blocked, and traffic is switched to the next device in the MSTP network in a forwarding state and a matching key. Each device must be configured to pass the traffic on the matching VLAN ID/mstp-instance.

[Table 32](#) defines the **mstp-instance** profile parameter. The **mstp-instance** profile binds an STP instance to a VLAN ID.

Table 32: mstp-instance profile

Parameter	Description
mstpName	A name for this MSTP instance and VLAN ID.

The stp-bind profile

The **stp-bind** profile is a system generated profile created when the **stp-bridge add** command is entered.

```
zSH> list stp-bind
stp-bind ethernet2/linegroup/1
stp-bind ethernet3/linegroup/2
2 entries found.

zSH> get stp-bind ethernet2/linegroup/1
stp-bind ethernet2/linegroup/1
portPriority: -> {176}
```

Table 33: stp-bind profile

Parameter	Description
portPriority	Used to specify the STP priority of this port.

The STP bridges

The **stp-bridge add** command is used to configure the bridges in the MSTP network ring. See [MSTP ring configuration on page 395](#).

MSTP ring configuration

This section describes the tasks to perform on each device in the MSTP ring:

- [Configuring the stp-params 0 profile, page 396](#)
- [Configuring mstp-instance profiles, page 396](#)
- [Configuring the MSTP network bridges, page 399](#)

Configuring the stp-params 0 profile

You *must* configure the **stp-params 0** file exactly the same on each device in the MSPT network. Each **stp-bridge add** command references the **stp-params 0** profile.

The **stp-params 0** profile *must* be configured on each device before proceeding with the **stp-bridge add** command.

- 1 Select and enter the **name** parameter, and set the **bridgePriority** to a multiple of 4096, and set the **forceVersion** parameter to 3 for MSTP.

```
zSH> update stp-params 0
stp-params 0
Please provide the following: [q]uit.
name: -----> {}: Region1
revision: -----> {0}:
bridgePriority: -> {36000}: 36864
forceVersion: ---> {2}: 3 <----- Must be configured 3 for MSTP.
fwdDelay: -----> {15}:
helloTime: -----> {2}:
migrateTime: ----> {3}:
txHoldCount: ----> {3}:
maxAge: -----> {20}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

- 2 Verify the changes.

```
zSH> get stp-params 0
stp-params 0
name: -----> {Region1}
revision: -----> {0}
bridgePriority: -> {36864}
forceVersion: ---> {3}
fwdDelay: -----> {15}
helloTime: -----> {2}
migrateTime: ----> {3}
txHoldCount: ----> {3}
maxAge: -----> {20}
```

Configuring mstp-instance profiles

After designing the MSTP network, create **mstp-instance** profiles on each device in the MSTP network to associate an instance to a VLAN ID.

All of the devices in the MSTP network *must* have matching **mstp-instance** profiles for the MSTP network to pass traffic in the MSTP ring.

- 1 Create all of the **mstp-instance** profiles for instance 1 on the first node in the MSTP configuration. Associate each instance 1 with each VLAN ID in the MSTP network.

```
zSH> new mstp-instance 1/111
mstp-instance 1/111
```



```
Please provide the following: [q]uit.
mstpName: -> {}: 1/111
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

- 2 Create all of the **mstp-instance** profiles for instance 2 on the first node in the MSTP configuration. Associate each instance 2 with each VLAN ID in the MSTP network.

```
zSH> new mstp-instance 2/122
mstp-instance 2/122
Please provide the following: [q]uit.
mstpName: -> {}: 2/122
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

- 3 Verify the instances and the associated VLANs.

```
zSH> list mstp-instance
mstp-instance 1/111
mstp-instance 1/112
mstp-instance 1/113
mstp-instance 1/114
mstp-instance 1/115
mstp-instance 2/121
mstp-instance 2/122
mstp-instance 2/123
mstp-instance 2/124
mstp-instance 2/125
mstp-instance 1/116
mstp-instance 1/117
mstp-instance 1/119
mstp-instance 1/120
mstp-instance 2/126
mstp-instance 2/127
mstp-instance 2/128
mstp-instance 2/129
mstp-instance 2/130
mstp-instance 1/100
mstp-instance 1/101
mstp-instance 2/999
mstp-instance 1/118
mstp-instance 2/502
24 entries found.
```

Or view a single instance:

```
zSH> get mstp-instance 1/111
mstp-instance 1/111
mstpName: -> {1/111}
```

- 4 When you have completed creating the instances in all the nodes in your MSTP network, verify that the instances exactly match in all nodes. A sample MSTP ring configuration is shown in [Table 34](#).

Table 34: MSTP ring where all the VLAN/instances match

Node 1 in MSTP ring	Node 2 in MSPT ring	Node 3 in MSTP ring
zSH> list mstp-instance	zSH> list mstp-instance	zSH> list mstp-instance
mstp-instance 1/111	mstp-instance 1/111	mstp-instance 1/111
mstp-instance 1/112	mstp-instance 1/112	mstp-instance 1/112
mstp-instance 1/113	mstp-instance 1/113	mstp-instance 1/113
mstp-instance 1/114	mstp-instance 1/114	mstp-instance 1/114
mstp-instance 1/115	mstp-instance 1/115	mstp-instance 1/115
mstp-instance 2/121	mstp-instance 2/121	mstp-instance 2/121
mstp-instance 2/122	mstp-instance 2/122	mstp-instance 2/122
mstp-instance 2/123	mstp-instance 2/123	mstp-instance 2/123
mstp-instance 2/124	mstp-instance 2/124	mstp-instance 2/124
mstp-instance 2/125	mstp-instance 2/125	mstp-instance 2/125
mstp-instance 1/116	mstp-instance 1/116	mstp-instance 1/116
mstp-instance 1/117	mstp-instance 1/117	mstp-instance 1/117
mstp-instance 1/119	mstp-instance 1/119	mstp-instance 1/119
mstp-instance 1/120	mstp-instance 1/120	mstp-instance 1/120
mstp-instance 2/126	mstp-instance 2/126	mstp-instance 2/126
mstp-instance 2/127	mstp-instance 2/127	mstp-instance 2/127
mstp-instance 2/128	mstp-instance 2/128	mstp-instance 2/128
mstp-instance 2/129	mstp-instance 2/129	mstp-instance 2/129
mstp-instance 2/130	mstp-instance 2/130	mstp-instance 2/130
mstp-instance 1/100	mstp-instance 1/100	mstp-instance 1/100
mstp-instance 1/101	mstp-instance 1/101	mstp-instance 1/101
mstp-instance 2/999	mstp-instance 2/999	mstp-instance 2/999
mstp-instance 1/118	mstp-instance 1/118	mstp-instance 1/118
mstp-instance 2/502	mstp-instance 2/502	mstp-instance 2/502
24 entries found.	24 entries found.	24 entries found.

Deleting a mstp-instance profile

When necessary, you can delete the MSTP instances.

Delete a **mstp-instance** profile.

```
zSH> delete mstp-instance 1/111
mstp-instance 1/111
1 entry found.
Delete mstp-instance 1/111? [y]es, [n]o, [q]uit : yes
```

mstp-instance 1/111 deleted.

Configuring the MSTP network bridges

shows a typical MSTP ring with traffic passing normally. In an MSTP ring functioning normally one port in the ring will be discarding and traffic does not pass in either direction.

When the VLAN ID is linked to an instance, the instance sets the preferred path. However, when the bridge is configured on the network facing Ethernet port, *all* the instances on a port must be the same.

Valid bridge types for MSTP rings are *rlink* and *tls*.

- 1 Configure the bridges for the MSTP ring on the first Ethernet port for instance 1. Each VLAN on this port will have instance 1 regardless of how the VLAN was linked in the **mstp-instance** profile.

The mechanism for setting MSTP port priority occurs the first time the port and VLAN ID are configured in the MSTP bridge configuration.

```
zSH> stp-bridge add 1-1-2-0/eth rlink vlan 0 slan 502 instance 1 staggered
Adding bridge on 1-1-2-0/eth
Created bridge-interface-record 1-1-2-0-eth-0/bridge
Bridge-path added successfully
Bridge-path added successfully
```

Verify the first bridge. The following shows the different states the bridge cycles through in an MSTP ring.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
rlk          ST    0/502    1/1/2/0/eth  1-1-2-0-eth-0-502/bridge  DIS  STP: DSNT
1 Bridge Interfaces displayed

zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
rlk          ST    0/502    1/1/2/0/eth  1-1-2-0-eth-0-502/bridge  LRN  STP: DSNT
1 Bridge Interfaces displayed

zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
rlk          ST    0/502    1/1/2/0/eth  1-1-2-0-eth-0/bridge  FWD S SLAN 502 VLAN 0 Intralink
STP: DSNT
1 Bridge Interfaces displayed
```

- 2 Create the rest of the bridge topology on the first Ethernet port of your configuration using all of the VLAN IDs in the MSTP configuration for instance 1.

```

zSH> stp-bridge add 1-1-2-0/eth rlink vlan 999 instance 1 tagged
Adding bridge on 1-1-2-0/eth
Created bridge-interface-record 1-1-2-0-eth-999/bridge
Bridge-path added successfully
Bridge-path added successfully

```

```

zSH> stp-bridge add 1-1-2-0/eth t1s vlan 100 instance 1 tagged
Adding bridge on 1-1-2-0/eth
Created bridge-interface-record 1-1-2-0-eth-100/bridge
Bridge-path added successfully

```

```

zSH> stp-bridge add 1-1-2-0/eth t1s vlan 101 instance 1 tagged
Adding bridge on 1-1-2-0/eth
Created bridge-interface-record 1-1-2-0-eth-101/bridge
Bridge-path added successfully

```

```

zSH> stp-bridge add 1-1-2-0/eth t1s vlan 111 instance 1 tagged
Adding bridge on 1-1-2-0/eth
Created bridge-interface-record 1-1-2-0-eth-111/bridge
Bridge-path added successfully

```

Continue until all of the MSTP bridges for instance 1 are configured.

View the bridges created for instance 1 on 1-1-2-0/eth uplink port of the MSTP network topology.

```
zSH> bridge show 1-1-2-0/eth
```

Orig							
Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
rlk	Tg	0/502	1/1/2/0/eth	1-1-2-0-eth-0/bridge	DIS	STP:	ALT
t1s	Tagged	100	1/1/2/0/eth	1-1-2-0-eth-100/bridge	DIS	STP:	ALT
t1s	Tagged	101	1/1/2/0/eth	1-1-2-0-eth-101/bridge	DIS	STP:	ALT
t1s	Tagged	111	1/1/2/0/eth	1-1-2-0-eth-111/bridge	DIS	STP:	ALT
t1s	Tagged	112	1/1/2/0/eth	1-1-2-0-eth-112/bridge	DIS	STP:	ALT
t1s	Tagged	113	1/1/2/0/eth	1-1-2-0-eth-113/bridge	DIS	STP:	ALT
t1s	Tagged	114	1/1/2/0/eth	1-1-2-0-eth-114/bridge	DIS	STP:	ALT
t1s	Tagged	115	1/1/2/0/eth	1-1-2-0-eth-115/bridge	DIS	STP:	ALT
t1s	Tagged	116	1/1/2/0/eth	1-1-2-0-eth-116/bridge	DIS	STP:	ALT
t1s	Tagged	117	1/1/2/0/eth	1-1-2-0-eth-117/bridge	DIS	STP:	ALT
t1s	Tagged	118	1/1/2/0/eth	1-1-2-0-eth-118/bridge	DIS	STP:	ALT
t1s	Tagged	119	1/1/2/0/eth	1-1-2-0-eth-119/bridge	DIS	STP:	ALT
t1s	Tagged	120	1/1/2/0/eth	1-1-2-0-eth-120/bridge	DIS	STP:	ALT
t1s	Tagged	121	1/1/2/0/eth	1-1-2-0-eth-121/bridge	DIS	STP:	ALT
t1s	Tagged	122	1/1/2/0/eth	1-1-2-0-eth-122/bridge	DIS	STP:	ALT
t1s	Tagged	123	1/1/2/0/eth	1-1-2-0-eth-123/bridge	DIS	STP:	ALT
t1s	Tagged	124	1/1/2/0/eth	1-1-2-0-eth-124/bridge	DIS	STP:	ALT
t1s	Tagged	125	1/1/2/0/eth	1-1-2-0-eth-125/bridge	DIS	STP:	ALT
t1s	Tagged	126	1/1/2/0/eth	1-1-2-0-eth-126/bridge	DIS	STP:	ALT
t1s	Tagged	127	1/1/2/0/eth	1-1-2-0-eth-127/bridge	DIS	STP:	ALT
t1s	Tagged	128	1/1/2/0/eth	1-1-2-0-eth-128/bridge	DIS	STP:	ALT
t1s	Tagged	129	1/1/2/0/eth	1-1-2-0-eth-129/bridge	DIS	STP:	ALT
t1s	Tagged	130	1/1/2/0/eth	1-1-2-0-eth-130/bridge	DIS	STP:	ALT
rlk	Tagged	999	1/1/2/0/eth	1-1-2-0-eth-999/bridge	DIS	STP:	ALT

24 Bridge Interfaces displayed

- 3 Create the rest of the bridge topology on the second Ethernet port of your configuration using all of the VLAN IDs in the MSTP configuration for instance 2.

```
zSH> stp-bridge add 1-1-3-0/eth rlink vlan 999 instance 2 tagged
Adding bridge on 1-1-3-0/eth
Created bridge-interface-record 1-1-3-0-eth-999/bridge
Bridge-path added successfully
Bridge-path added successfully
```

```
zSH> stp-bridge add 1-1-3-0/eth tls vlan 100 instance 2 tagged
Adding bridge on 1-1-3-0/eth
Created bridge-interface-record 1-1-3-0-eth-100/bridge
Bridge-path added successfully
```

```
zSH> stp-bridge add 1-1-3-0/eth tls vlan 101 instance 2 tagged
Adding bridge on 1-1-3-0/eth
Created bridge-interface-record 1-1-3-0-eth-101/bridge
Bridge-path added successfully
```

```
zSH> stp-bridge add 1-1-3-0/eth tls vlan 111 instance 2 tagged
Adding bridge on 1-1-3-0/eth
Created bridge-interface-record 1-1-3-0-eth-111/bridge
Bridge-path added successfully
```

View the bridges created for instance 2 on *1-1-3-0/eth* uplink port of the MSTP network topology.

```
zSH> bridge show 1-1-3-0/eth
```

Orig							
Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
rlk		ST 0/502	1/1/3/0/eth	1-1-3-0-eth-0-502/bridge	FWD	S	SLAN 502 VLAN 0 default
STP: ROOT							
tls	Tagged	100	1/1/3/0/eth	1-1-3-0-eth-100/bridge	FWD	STP:	ROOT
tls	Tagged	101	1/1/3/0/eth	1-1-3-0-eth-101/bridge	FWD	STP:	ROOT
tls	Tagged	111	1/1/3/0/eth	1-1-3-0-eth-111/bridge	FWD	STP:	ROOT
tls	Tagged	112	1/1/3/0/eth	1-1-3-0-eth-112/bridge	FWD	STP:	ROOT
tls	Tagged	113	1/1/3/0/eth	1-1-3-0-eth-113/bridge	FWD	STP:	ROOT
tls	Tagged	114	1/1/3/0/eth	1-1-3-0-eth-114/bridge	FWD	STP:	ROOT
tls	Tagged	115	1/1/3/0/eth	1-1-3-0-eth-115/bridge	FWD	STP:	ROOT
tls	Tagged	116	1/1/3/0/eth	1-1-3-0-eth-116/bridge	FWD	STP:	ROOT
tls	Tagged	117	1/1/3/0/eth	1-1-3-0-eth-117/bridge	FWD	STP:	ROOT
tls	Tagged	118	1/1/3/0/eth	1-1-3-0-eth-118/bridge	FWD	STP:	ROOT
tls	Tagged	119	1/1/3/0/eth	1-1-3-0-eth-119/bridge	FWD	STP:	ROOT
tls	Tagged	120	1/1/3/0/eth	1-1-3-0-eth-120/bridge	FWD	STP:	ROOT
tls	Tagged	121	1/1/3/0/eth	1-1-3-0-eth-121/bridge	FWD	STP:	ROOT
tls	Tagged	122	1/1/3/0/eth	1-1-3-0-eth-122/bridge	FWD	STP:	ROOT
tls	Tagged	123	1/1/3/0/eth	1-1-3-0-eth-123/bridge	FWD	STP:	ROOT
tls	Tagged	124	1/1/3/0/eth	1-1-3-0-eth-124/bridge	FWD	STP:	ROOT
tls	Tagged	125	1/1/3/0/eth	1-1-3-0-eth-125/bridge	FWD	STP:	ROOT
tls	Tagged	126	1/1/3/0/eth	1-1-3-0-eth-126/bridge	FWD	STP:	ROOT
tls	Tagged	127	1/1/3/0/eth	1-1-3-0-eth-127/bridge	FWD	STP:	ROOT
tls	Tagged	128	1/1/3/0/eth	1-1-3-0-eth-128/bridge	FWD	STP:	ROOT
tls	Tagged	129	1/1/3/0/eth	1-1-3-0-eth-129/bridge	FWD	STP:	ROOT
tls	Tagged	130	1/1/3/0/eth	1-1-3-0-eth-130/bridge	FWD	STP:	ROOT
rlk	Tagged	999	1/1/3/0/eth	1-1-3-0-eth-999/bridge	FWD	S	VLAN 999 default STP: ROOT

24 Bridge Interfaces displayed

- 4 Configure each node in the MSTP ring with the identical VLAN, instance 1 and instance 2 configurations.

Bridge configurations for VLAN ID and instance 1, VLAN ID and instance 2 must be identical. However, the two port numbers on the device do not need to match across devices.

MSTP ring operation

This section describes how a simple MSTP ring functions:

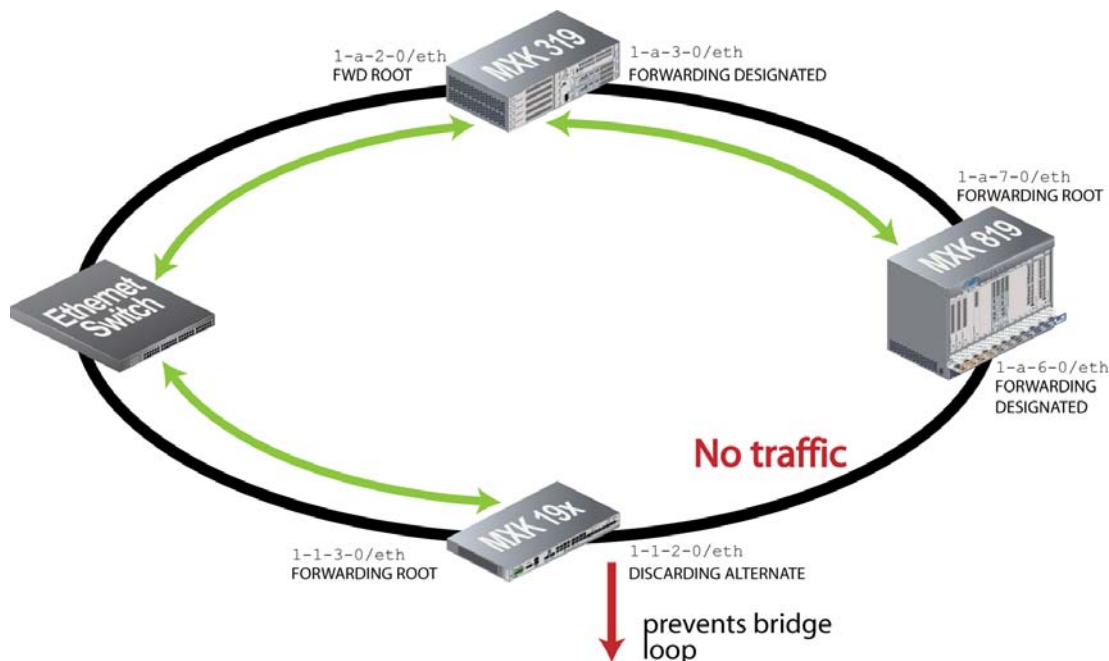
- [MSTP ring normal operation, page 402](#)
- [MSTP ring with blocked port on the MXK 819, page 404](#)

MSTP ring normal operation

This MSTP ring consists of one MKK-194/198, one MXK 319, one MXK 819, and one Ethernet router.

In order for an MSTP ring to efficiently pass traffic, one link in the loop *must* not pass traffic either due to a DISCARDING port as shown in [Figure 53](#) (1-1-2-0/eth), or due to a BLOCKED port, as shown in [Figure 54](#).

Figure 53: Example MSTP ring normal traffic



Node 1: MXK-194/198 states as shown in [Figure 53](#).

MSTP bridge VLAN 100 on 1-1-3-0/eth is ROOT and FORWARDING.

Bridge VLAN 100 on 1-1-2-0/eth is DISCARDING and ALTERNATE. Traffic cannot pass from the MXK 19x to the MXK 819.

```
zSH> bridge show vlan 100
```

```
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
tls             Tagged 100    1/1/3/0/eth  1-1-3-0-eth-100/bridge  FWD  STP: ROOT
tls             Tagged 100    1/1/2/0/eth  1-1-2-0-eth-100/bridge  DIS  STP: ALT
2 Bridge Interfaces displayed
```

Bridges on *1-1-2-0/eth* are DISCARDING and ALTERNATE. Traffic cannot pass from the MXK 19x to the MXK 819.

```
zSH> bridge show
```

```
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
rlk             Tg    0/502    1/1/2/0/eth  1-1-2-0-eth-0/bridge    DIS  STP: ALT
tls             Tagged 100    1/1/2/0/eth  1-1-2-0-eth-100/bridge  DIS  STP: ALT
2 Bridge Interfaces displayed
```

Node 2: MXK 819 states as shown in [Figure 53](#).

MSTP bridges on *1-a-7-0/eth* are FORWARDING and ROOT.

```
zSH> bridge show
```

```
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
rlk             ST    0/502    1/a/7/0/eth  ethernet7-0-502/bridge  FWD S SLAN 502 VLAN 0 default STP: ROOT
tls             Tagged 100    1/a/7/0/eth  ethernet7-100/bridge    FWD  STP: ROOT
2 Bridge Interfaces displayed
```

MSTP bridges on *1-a-6-0/eth* are FORWARDING and DISIGNATED, however traffic is discarded on the MXK 19x to prevent bridge looping.

```
zSH> bridge show
```

```
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
rlk             ST    0/502    1/a/6/0/eth  ethernet6-0-502/bridge  FWD S SLAN 502 VLAN 0 Intralink STP: DSNT
tls             Tagged 100    1/a/6/0/eth  ethernet6-100/bridge    FWD  STP: DSNT
2 Bridge Interfaces displayed
```

Node 3: MXK 319 states as shown in [Figure 53](#).

MSTP bridges on *1-a-2-0/eth* are FORWARDING and ROOT.

```
zSH> bridge show
```

```
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
rlk             ST    0/502    1/a/2/0/eth  ethernet2-0-502/bridge  FWD S SLAN 502 VLAN 0 default STP: ROOT
tls             Tagged 100    1/a/2/0/eth  ethernet2-100/bridge    FWD  STP: ROOT
2 Bridge Interfaces displayed
```

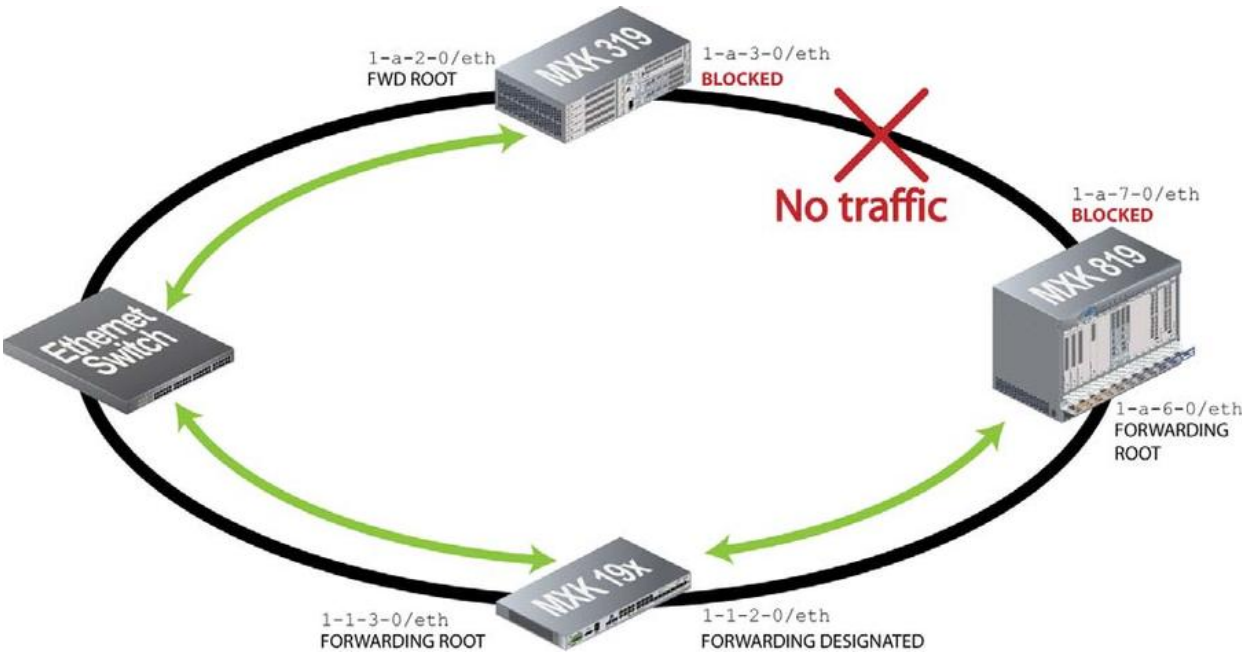
MSTP bridges on *1-a-3-0/eth* are FORWARDING AND DESIGNATED.

```
zSH> bridge show
```

Orig							
Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
rlk	ST	0/502	1/a/3/0/eth	ethernet3-0-502/bridge	FWD	S	SLAN 502 VLAN 0 Intralink STP: DSNT
tls	Tagged	100	1/a/3/0/eth	ethernet3-100/bridge	FWD	STP: DSNT	
2 Bridge Interfaces displayed							

MSTP ring with blocked port on the MXK 819

Figure 54: MSTP ring with blocked port on the MXK 819



Node 1: MXK 19x states as shown ins shown in Figure 54.

In this example, when a port on the MXK 819 goes down, the states of MSTP bridges on 1-1-2-0/eth change to FORWARDING DESIGNATED since traffic is now BLOCKED elsewhere on the MSTP ring.

```
zSH> bridge show
```

Orig							
Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
rlk	Tg	0/502	1/1/2/0/eth	1-1-2-0-eth-0/bridge	FWD	STP: DSNT	
tls	Tagged	100	1/1/2/0/eth	1-1-2-0-eth-100/bridge	FWD	STP: DSNT	
2 Bridge Interfaces displayed							

The state of the bridges on 1-1-3-0/eth remain FORWARDING ROOT.

```
zSH> bridge show
```

Orig							
Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
rlk	ST	0/502	1/1/3/0/eth	1-1-3-0-eth-0-502/bridge	FWD	S	SLAN 502 VLAN 0 default STP: ROOT
tls	Tagged	100	1/1/3/0/eth	1-1-3-0-eth-100/bridge	FWD	STP: ROOT	
2 Bridge Interfaces displayed							

Node 2: The MXK 819 states as shown in [Figure 54](#).

Port *1-a-7-0/eth* goes down changing the state to BLOCKED.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
rlk           ST    0/502  1/a/7/0/eth      ethernet7-0-502/bridge  BLK  A 00:00:00:00:00:00
tls           Tagged 100  1/a/7/0/eth      ethernet7-100/bridge   BLK  A 00:00:00:00:00:00
2 Bridge Interfaces displayed
```

Port *1-a-6-0/eth* changes to FORWARDING ROOT and traffic can now pass between the MXK 819 and the MXK 19x.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
rlk           ST    0/502  1/a/6/0/eth      ethernet6-0-502/bridge  FWD S SLAN 502 VLAN 0 Intralink STP: ROOT
tls           Tagged 100  1/a/6/0/eth      ethernet6-100/bridge   FWD  STP: ROOT
2 Bridge Interfaces displayed
```

Node 3: The MXK 319 states as shown in [Figure 54](#).

When port 7 on the MXK 819 goes down port 3 on the MXK 319 goes down as well. Traffic does not pass on this link and bridge looping is prevented.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
rlk           ST    0/502  1/a/3/0/eth      ethernet3-0-502/bridge  BLK  A 00:00:00:00:00:00
tls           Tagged 100  1/a/3/0/eth      ethernet3-100/bridge   BLK  A 00:00:00:00:00:00
2 Bridge Interfaces displayed
```

Traffic is passed to the MSTP ring through port 2 which remains in a FORWARD ROOT state.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
rlk           ST    0/502  1/a/2/0/eth      ethernet2-0-502/bridge  FWD S SLAN 502 VLAN 0 default STP: ROOT
tls           Tagged 100  1/a/2/0/eth      ethernet2-100/bridge   FWD  STP: ROOT
2 Bridge Interfaces displayed
```

MSTP ring IP on a bridge in-band device management

Because there are two paths off the devices in an MSTP ring, Zhone recommends configuring IP on a bridge for in-band management.

For additional information on IP on a bridge for device management, see [In-band management on the MXK on page 49](#)

Configuring IP on a bridge on a MSTP device

When configuring IP on a bridge for a MSTP ring, you must use a VLAN ID in use by a STP bridge in the MSTP network.

- 1 View the STP bridges on the device to see which existing bridges and VLAN IDs are used in the MSTP ring.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
rlk      ST    0/502    1/a/6/0/eth      ethernet6-0-502/bridge  FWD S SLAN 502 VLAN 0 Intralink
STP: DSNT
tls      Tagged 100    1/a/6/0/eth      ethernet6-100/bridge   FWD STP: DSNT
tls      Tagged 101    1/a/6/0/eth      ethernet6-101/bridge   FWD STP: DSNT
```

Since a TLS bridge already exists on the device, an additional bridge does not need to be created.

- 2 Enter **interface add interface/type** with the type as *ipobridge* and the VLAN ID from an existing RSTP TLS bridge.

```
zSH> interface add 1-a-6-0/ipobridge vlan 100 192.168.8.21/24
Created ip-interface-record ipobridge-100/ip.
```



Note: Ipv4 is required for all IP termination on the MXK, including **ipobridge** interfaces. IPv6 is not supported for IP termination on the MXK.

- 3 Verify the interface.

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 10.54.1.111/24    00:01:47:22:99:f8   ethernet1
1/a/6/0/ip     UP      1 192.168.8.21/24   00:01:47:2a:3b:e8   ipobridge-100
```

- 4 Verify the STP bridges and the IP on a bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
rlk      ST    0/502    1/a/6/0/eth      ethernet6-0-502/bridge  FWD S SLAN 502 VLAN 0 Intralink
STP: DSNT
tls      Tagged 100    1/a/6/0/eth      ethernet6-100/bridge   FWD STP: DSNT
tls      Tagged 100    1/a/6/0/ipobridge ipobridge-100/bridge   UP  D 00:01:47:2a:3b:e8
                                     D 192.168.8.21
```

Shaping Traffic: Class of Service Queuing

Class of Service (CoS) queuing controls traffic to optimize or guarantee performance. This shaping of traffic generally exists to increase bandwidth so

you can get more throughput to a device, or to decrease latency, so you do not have jitter in time sensitive data streams as in voice or video.

Congestion happens for various reasons. If you have a higher bandwidth line feeding into a smaller bandwidth line, or if you have multiple similar size lines feeding into a single line. Both of these can be considered feeding too much data (a big pipe) into a small pipe.

Queuing defines which VLAN will be able to use how much of the physical interface.

The MXK supports setting CoS values in Ethernet VLAN headers for bridged packets. This service enables you to assign a service level or CoS to an Ethernet VLAN interface that is transported across a uplink, intralink, or downlinked tagged bridge. The configured CoS level specifies the packet priority and queueing methods used to transport the packet through the Ethernet network. The MXK sets and preserves the CoS settings to ensure these settings are passed to other Ethernet devices in the network for QoS processing.



Note: Statistics on demand must be enabled on bridges configured for CoS. See [Bridge statistics-on-demand on page 416](#) for more information.

CoS values range from 0 — 7, with the lowest priority being 0 and the highest priority 7. The MXK supports eight queues per physical interface meaning that frames with a 0 CoS value are put into queue number 0; frames with a 1 CoS value are put into queue number 1, and so forth.

These are strict priority queues which mean that everything is cleared out of the high priority queue first. Only after that queue is empty is the next queue serviced. Since these are strict priority queues it is possible that the lower priority queues may get overloaded while the higher priority queues are being cleared.

Frames which require the highest throughput or are sensitive to latency (the amount of time between received packets) should be in higher priority queues. Since queuing is relative to the type of traffic, the priority settings depend on the type of traffic. Normally video and voice are more sensitive to throughput and latency issues.

Where CoS queuing takes place is dependent on the cards involved. GPON and Active Ethernet cards have queuing performed on the line card. For ADSL the queuing takes place on the uplink card.

Configuring Class of Service

The following parameters in the bridge interface record are used for Ethernet COS support.

Table 35: COS parameters in the bridge-interface-record profile

Parameter	Description
vlanIdCOS	Specifies the value loaded into the COS field of the VLAN header when an untagged packet received on this interface is tagged (VLAN ID inserted) for bridging. Value range is 0 to 7. Default is 0.
outgoingCOSOption	Specifies whether to insert the VLAN COS bits on packets bridged through this interface. Values: Disable Leave any existing COS values unchanged. This is the default value. All Replace the current COS values in all VLAN headers in tagged and untagged packets originating and transported through this device.
outgoingCOSValue	For outgoing tagged packets, specifies the value used to overwrite any existing COS value in the VLAN header. Value range is 0 to 7. Default is 0.

To display the **bridge-interface-record** profile, enter the **show bridge-interface-record** command.

```

zSH> show bridge-interface-record
vpi:-----> {0}
vci:-----> {0}
vlanId:-----> {0 - 2147483647}
stripAndInsert:-----> false true
customARP:-----> false true
filterBroadcast:-----> false true
learnIp:-----> false true
learnUnicast:-----> false true
maxUnicast:-----> {0 - 2147483647}
learnMulticast:-----> false true
forwardToUnicast:-----> false true
forwardToMulticast:-----> false true
forwardToDefault:-----> false true
bridgeIfCustomDHCP:-----> false true
bridgeIfConfigGroupIndex:-> {0 - 2147483647}
vlanIdCOS:-----> {0 - 7}
outgoingCOSOption:-----> disable all
outgoingCOSValue:-----> {0 - 7}

```

Adding a bridge with a CoS value

This example adds interface *1-13-1-0/eth* with a `vlanIDCOS` value of 7. This value is inserted into the priority field of the VLAN header when an untagged packet received on this interface is tagged (VLAN ID inserted) for bridging.

```

zSH> bridge add 1-13-1-0/eth downlink vlan 100 tagged COS 7
Adding bridge on 1-13-1-0/eth

```

```
Created bridge-interface-record 1-13-1-0-eth-100/bridge
```

This example adds interface *1-13-2-0/eth* with a **vlanIdCOS** value of 7 and enables the overwriting of the VLAN ID in all outgoing packets with the value of 7.

```
zSH> bridge add 1-13-2-0/eth downlink vlan 100 tagged COS 7 outCOSall 7
Adding bridge on 1-13-2-0/eth
Created bridge-interface-record 1-13-2-0-eth-100/bridge
```

COS and SCOS replacement on Ethernet frames

COS or SCOS replacement is the ability to overwrite COS or SCOS in Ethernet frames for traffic entering Active Ethernet line cards and sending the configured COS value to the uplink when the **incomingCOSOption** or the **s_tagIncomingCOSOption** parameters are enabled in the **bridge-interface-record** profile.

COS and SCOS replacement on Ethernet frames is available on the following Active Ethernet single-slot line cards:

- MXK-AE-2X10G-8XGE
- MXK-AEX20-FE/GE-CSFP
- MXK-AEX20-FE/GE

The current priority levels set for **vlanIdCOS** and **s-tagIdCOS** parameters are 0 - 7 where 0: disable and 7: highest priority. When COS or SCOS values are invoked with **incomingCOSOption** or **s_tagIncomingCOSOption** set to *all*, the priority levels will be 0: lowest priority and 7: highest priority.

Enabling COS replacement on a Ethernet downlink bridge

Enable COS replacement with the **bridge add interface/type downlink vlanID tagged incosall cos value** command.

Entering the *incosall* key word with the **bridge add** command enables the **incomingCOSOption** parameter in the **bridge-interface-record** profile and places the COS value in the **vlanIdCOS** parameter. The COS value found in the **vlanIdCOS** parameter overwrites any existing COS value in the incoming packet before sending the packet to the uplink.

Configure the downlink bridge for COS replacement.

```
zSH> bridge add 1-6-1-0/eth downlink vlan 102 tagged incosall cos 7
Adding bridge on 1-6-1-0/eth
Created bridge-interface-record 1-6-1-0-eth-102/bridge
```

The bridge-interface-record shows the **vlanIdCOS** parameter is set to 7 and the **incomingCOSOption** parameter is set to all. The COS value of 7 will be inserted into all Ethernet packets with COS values.

```
zSH> get bridge-interface-record 1-6-1-0-eth-102/bridge
bridge-interface-record 1-6-1-0-eth-102/bridge
```

```

vpi: -----> {0}
vci: -----> {0}
vlanId: -----> {102}
stripAndInsert: -----> {false}
customARP: -----> {false}
filterBroadcast: -----> {false}
learnIp: -----> {true}
learnUnicast: -----> {true}
maxUnicast: -----> {5}
learnMulticast: -----> {true}
forwardToUnicast: -----> {false}
forwardToMulticast: -----> {false}
forwardToDefault: -----> {true}
bridgeIfCustomDHCP: -----> {false}
bridgeIfIngressPacketRuleGroupIndex: -> {0}
vlanIdCOS: -----> {7}
outgoingCOSOption: -----> {disable}
outgoingCOSValue: -----> {0}
s-tagTPID: -----> {0x8100}
s-tagId: -----> {0}
s-tagStripAndInsert: -----> {true}
s-tagOutgoingCOSOption: -----> {s-tagdisable}
s-tagIdCOS: -----> {0}
s-tagOutgoingCOSValue: -----> {0}
mcastControlList: -----> {}
maxVideoStreams: -----> {0}
isPPPoA: -----> {false}
floodUnknown: -----> {false}
floodMulticast: -----> {false}
bridgeIfEgressPacketRuleGroupIndex: --> {0}
bridgeIfTableBasedFilter: -----> {NONE(0)}
bridgeIfDhcpLearn: -----> {NONE(0)}
mvrVlan: -----> {0}
vlan-xlate-from: -----> {0}
slan-xlate-from: -----> {0}
bridge-type: -----> {downlink}
incomingCOSOption: -----> {all}
s_tagIncomingCOSOption: -----> {disable}

```

Enabling SCOS replacement on a Ethernet downlink bridge

Enable SCOS replacement with the **bridge add interface/type downlink vlanID slanID staged sincosall scos value** command.

Entering the *sincosall* key word with the **bridge add** command enables the **s_tagIncomingCOSOption** parameter in the **bridge-interface-record** profile and places the SCOS value in the **s-tagIdCOS** parameter. The SCOS value found in the **s-tagIdCOS** parameter overwrites any existing SCOS value in the incoming packet before sending the packet to the uplink.

Configure the downlink bridge for SCOS replacement.

```

zSH> bridge add 1-6-1-0/eth downlink vlan 203 slan 303 staged sincosall scos 7
Adding bridge on 1-6-1-0/eth

```

Created bridge-interface-record 1-6-1-0-eth-203-303/bridge

The bridge-interface-record shows the **s-tagIdCOS** parameter is set to 7 and the **s_tagIncomingCOSOption** parameter is set to all. The COS value of 7 will be inserted into all Ethernet packets with COS values.

```
zSH> get bridge-interface-record 1-6-1-0-eth-203-303/bridge
bridge-interface-record 1-6-1-0-eth-203-303/bridge
vpi: -----> {0}
vci: -----> {0}
vlanId: -----> {203}
stripAndInsert: -----> {false}
customARP: -----> {false}
filterBroadcast: -----> {false}
learnIp: -----> {true}
learnUnicast: -----> {true}
maxUnicast: -----> {5}
learnMulticast: -----> {true}
forwardToUnicast: -----> {false}
forwardToMulticast: -----> {false}
forwardToDefault: -----> {true}
bridgeIfCustomDHCP: -----> {false}
bridgeIfIngressPacketRuleGroupIndex: -> {0}
vlanIdCOS: -----> {0}
outgoingCOSOption: -----> {disable}
outgoingCOSValue: -----> {0}
s-tagTPID: -----> {0x8100}
s-tagId: -----> {303}
s-tagStripAndInsert: -----> {false}
s-tagOutgoingCOSOption: -----> {s-tagdisable}
s-tagIdCOS: -----> {7}
s-tagOutgoingCOSValue: -----> {0}
mcastControlList: -----> {}
maxVideoStreams: -----> {0}
isPPPoA: -----> {false}
floodUnknown: -----> {false}
floodMulticast: -----> {false}
bridgeIfEgressPacketRuleGroupIndex: --> {0}
bridgeIfTableBasedFilter: -----> {NONE(0)}
bridgeIfDhcpLearn: -----> {NONE(0)}
mvrVlan: -----> {0}
vlan-xlate-from: -----> {0}
slan-xlate-from: -----> {0}
bridge-type: -----> {downlink}
incomingCOSOption: -----> {disable}
s_tagIncomingCOSOption: -----> {all}
```

“Denial of Service” prevention

Enhanced broadcast storm protection the line cards prevents upstream broadcast storms. Broadcasts received into the system are placed in the lowest priority queue for exception packets. This queue is limited to 1,000 packets

per second, the maximum number the hardware will allow onto the exception path. This throttling mitigates broadcast storms.

Bridging differences between the MALC and MXK

The MALC and the SLMS devices which have a similar architecture — the MALC XP, Raptor XP, and EtherXtend 34xx — have behaviors which are different than the MXK. The MXK processes one tag at a time. If double tags are present, the MXK processes based on the outer tag (*stag*) only. Because only the outer tag is processed, an Ethernet frame with an SLAN 200 and VLAN 75 only forwards the frame based on the SLAN 200.

MXK bridge statistics-on-demand

Statistics are enabled by default on the ingress and egress of ADSL, VDSL, T1 bonded, EFM cards, and the egress of GPON cards. Statistics-on-demand is enabled on certain Ethernet uplink cards, Active Ethernet cards and GPON cards on the ingress, see [MXK bridge statistics-on-demand on page 413](#).

Bridge statistics are also enabled on all *ipobridge* interfaces.

The uplink cards that support statistics-on-demand are:

- MXK-UPLINK-2X10G-8X1G-TOP
- MXK-UPLINK-2X10G-8X1G-CLK
- MXK-UPLINK-6X1G-CLK

Due to hardware limitations, the following cards support bytes only on the ingress. Counters are displayed in packets on the egress:

- MXK-UPLINK-2X10G-8X1GE
- MXK-UPLINK-8X1GE
- MXK-AEX20-FE/GE-2S
- MXK-AEX20-FE/GE

Bridge interface statistics-on-demand overview

There are two commands for viewing statistics on bridges. The first command, **bridge stats**, displays all of the packet counters that have passed through the interface. The second command, **bridge rates**, displays all of the packets that pass through the bridge interface in rate-per-second.

The **bridge stats** command can display statistics for all bridge interfaces that display statistics, for a specified bridge interface, or for bridges on a specified VLAN ID.

The **bridge stats** command displays both received and transmitted packet counters for the *ipobridge* interface, transmitted packet counters for the GPON bridge interfaces, and blank traffic counters for the Ethernet bridge interfaces.

The default counters for the **bridge stats** command are packet counters. To display counters in bytes, byte counters must be enabled. When byte counters are enabled, packet counter will not be displayed. When statistics are enabled by default, as on ADSL, VDSL, T1 bonded, and EFM cards, they are not available to display byte counters.

Only those bridge interfaces that support statistics-on-demand with the **bridge stats enable interfaceName/bridge bytes** command can display statistical output in bytes.

zSH> bridge stats

Interface Name	Received Packets			Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
ethernet2-3101/bridge	--	--	--	--	--	--	--	0	0	0	0	--	--
ethernet2-998/bridge	--	--	--	--	--	--	--	0	0	0	0	--	--
ipobridge-3150/bridge	54148	5933	924	48249	0	0	0	0	0	0	0	--	--
ethernet4/bridge	--	--	--	--	--	--	--	0	0	0	0	--	--
ethernet2-3150/bridge	--	--	--	--	--	--	--	0	0	0	0	--	--
1-6-1-302-gponport-998/bridge	--	--	--	0	0	0	9307	0	0	0	0	--	--
1-6-1-402-gponport-3101/bridge	--	--	--	0	0	0	60671	0	0	0	0	--	--
1-6-1-301-gponport-998/bridge	--	--	--	0	5144	2321	0	0	0	0	0	--	--
1-6-1-401-gponport-3101/bridge	--	--	--	1564k	0	8187	0	0	0	0	0	--	--

9 Bridge Interfaces displayed

bridge statistics commands on bridge interfaces with statistics enabled by default

View bridge interface statistics that are enabled by default

Viewing bridge statistics enabled by default

Enter the **bridge stats** *interfaceName/bridge* command to view statistics on a bridge interface where statistics are enabled by default.

In this case, the bridge interface is ADSL.

zSH> bridge stats 1-1-29-0-adsl-0-35/bridge

Interface Name	Received Packets			Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
1-1-29-0-adsl-0-35/bridge	597	0	5	600	4	0	0	0	0	0	0	--	--

1 Bridge Interfaces displayed

Enter the **bridge rates** *interfaceName/bridge* command to view statistics in rate-per-second.

zSH> bridge rates 1-1-29-0-adsl-0-35/bridge

Interface Name	Received Packets/Sec			Transmitted Packets/Sec				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
1-1-29-0-adsl-0-35/bridge	1	0	1	1	1	0	0	0	0	0	0	--	--

1 Bridge Interfaces displayed

Viewing bridge statistics by ADSL port

In the case of ADSL bridges, traffic is separated by VCI and statistics for all bridges on the port are viewed with the **bridge stats** *shelf/slot/port/interfaceType* command.

zSH> bridge stats 1/1/30/0/adsl

Interface Name	Received Packets			Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
1-1-30-0-adsl-0-35/bridge	675	0	4	677	4	0	0	0	0	0	0	--	--
1-1-30-0-adsl-0-36/bridge	0	416	0	0	52	200	0	0	0	0	0	--	--

2 Bridge Interfaces displayed

Viewing bridge statistics by VLAN ID

Enter the **bridge stats vlanid** command to view bridge statistics by VLAN ID.

zSH> **bridge stats vlan 999**

Interface Name	Received Packets			Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	Bcast	Error	UCast	MCast	Bcast	Alarm	Received	Transmitted
ethernet5-999/bridge	0	67	0	0	0	0	0	0	0	0	0	--	--
1-b-5-0-eth-999/bridge	0	4	0	0	65	0	0	0	0	0	0	--	--
1-3-1-0-vdsl-0-38-999/bridge	0	0	0	0	68	0	0	0	0	0	0	--	--

3 Bridge Interfaces displayed

Use the bridge stats reset, clear, list, and rules commands for default and enabled statistics

Using the bridge statistics reset command

Use the **bridge statistics reset interfaceName/bridge** command to display and clear statistics and rates on bridge interfaces.

- 1 Enter the **bridge stats reset interfaceName/bridge** command to display and reset statistical counters to 0, and resume counting.

Bridge interface with statistics-on-demand enabled.

zSH> **bridge stats reset ethernet5-3605/bridge**

Interface Name	Received Packets			Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	Bcast	Error	UCast	MCast	Bcast	Alarm	Received	Transmitted
ethernet5-3605/bridge	23	0	2	15	137	699	0	0	0	0	0	--	--

1 Bridge Interfaces displayed
Bridge statistics cleared

Bridge interface with default statistics.

zSH> **bridge stats reset 1-1-26-0-adsl-0-35/bridge**

Interface Name	Received Packets			Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	Bcast	Error	UCast	MCast	Bcast	Alarm	Received	Transmitted
1-1-26-0-adsl-0-35/bridge	677	0	5	680	4	0	0	0	0	0	0	--	--

1 Bridge Interfaces displayed
Bridge statistics cleared

- 2 Enter the **bridge stats interfaceName/bridge** command immediately following the **bridge stats reset interfaceName/bridge** command to display counters reset to 0.

zSH> **bridge stats ethernet5-3605/bridge**

Interface Name	Received Packets			Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	Bcast	Error	UCast	MCast	Bcast	Alarm	Received	Transmitted
ethernet5-3605/bridge	0	0	0	0	0	0	0	0	0	0	0	--	--

1 Bridge Interfaces displayed

- 3 Enter the **bridge stats interfaceName/bridge** command after an interval to display the reset packet counter information.

zSH> **bridge stats ethernet5-3605/bridge**

Interface Name	Received Packets			Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	Bcast	Error	UCast	MCast	Bcast	Alarm	Received	Transmitted
ethernet5-3605/bridge	0	0	0	0	0	0	0	0	0	0	0	--	--

```
ethernet5-3605/bridge      4      0      1      4      46      213      0      0      0      0      0      0      --      --
1 Bridge Interfaces displayed
```

Entering the bridge statistics clear command

Enter the **bridge stats clear** *interfaceName/bridge* command to clear statistics and rates without displaying them.

```
zSH> bridge stats clear ethernet5-3605/bridge
Bridge statistics cleared
```

Displaying all bridge interfaces with enabled on-demand statistics

Use the **bridge stats list** command to view all bridge interfaces with enabled statistics-on-demand.

Enter the **bridge stats list** command.

```
zSH> bridge stats list
ethernet5-207/bridge
ethernet5-213/bridge
ethernet5-3605/bridge
1-b-5-0-eth-207/bridge
ethernet6-201/bridge
ethernet5-999/bridge
1-b-5-0-eth-999/bridge
7 bridges have on-demand stats enabled
```

Displaying statistics-on-demand rules

Use the **bridge stats rules** command to display the total number of on-demand rules in use and the rules remaining on a per slot basis.

Enter the **bridge stats rules** command.

```
zSH> bridge stats rules
Slot  Total Rules  Total Rules
      In Use      Remaining
====  =====
a          5          251
b          2          254
```

Bridge statistics-on-demand

Statistics are available on-demand for certain bridge types on the MXK.

You can enable or disable packet or byte counters with the **bridge stats enable|disable** command on a per port basis.

The following cards support statistics on-demand:

- uplink Ethernet cards
- active Ethernet line cards

- GPON line cards statistics-on-demand must be enabled for the ingress, the egress is enabled by default.

There are a total of 256 interfaces on which statistics can be enabled per port. Use the **bridge stats rules** command to view the total on-demand rules in use as well as rules remaining on the slot. See [Displaying statistics-on-demand rules on page 416](#).

Statistics-on-demand for bridge interface configuration

This section covers the following **bridge stats** command procedures:

- [Enabling statistics-on-demand, page 417](#)
- [Disabling statistics-on-demand, page 418](#)
- [Viewing bridge statistics-on-demand in rate-per-second, page 418](#)
- [Viewing bridge statistics-on-demand in bytes, page 418](#)
- [Viewing statistics-on-demand per VLAN ID, page 419](#)

View bridge statistics on Ethernet bridges

Ethernet bridges and the ingress of GPON bridges must be enabled to display statistics.

Enabling statistics-on-demand

After enabling the bridge for statistics-on-demand, the default for viewing statistics are in packet counters that are cumulative.

To view statistics in rate-per-second, enter the **bridge rates interfaceName/bridge** command.

To view statistics in bytes, enable the bridge interface using the **bridge stats enable interfaceName/bridge bytes** command.

- 1 View bridge statistics for an Ethernet bridge interface.

```
zSH> bridge stats ethernet5-3605/bridge
```

Interface Name	Received Packets			Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	Bcast	Error	UCast	MCast	Bcast	Alarm	Received	Transmitted
ethernet5-3605/bridge	--	--	--	--	--	--	--	0	0	0	0	--	--

```
1 Bridge Interfaces displayed
```

Since statistics on the Ethernet bridge interface have not been enabled, the statistics fields are empty.

- 2 Enable the specified Ethernet bridge interface to view statistics.

```
zSH> bridge stats enable ethernet5-3605/bridge
on-demand stats on interface "ethernet5-3605/bridge" have been enabled
```



Note: Bridge statistics are enabled to packets by default.

3 View statistics on the Ethernet bridge interface.



Note: The bridge interface statistics table may take up to several minutes for statistical data to completely display.

```
zSH> bridge stats ethernet5-3605/bridge
```

Interface Name	Received Packets				Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	Error	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
ethernet5-3605/bridge	7	0	1	0	7	64	339	0	0	0	0	0	--	--

1 Bridge Interfaces displayed

Disabling statistics-on-demand

Enter the **bridge stats disable interfaceName/bridge** command to disable statistics-on-demand on a bridge interface.

```
zSH> bridge stats disable ethernet5-3605/bridge
```

on-demand stats on interface "ethernet5-3605/bridge" have been disabled

Viewing bridge statistics-on-demand in rate-per-second

View statistics on the specified Ethernet bridge interface in rate-per-second.

```
zSH> bridge rates ethernet5-3605/bridge
```

Interface Name	Received Packets/Sec				Transmitted Packets/Sec				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	Error	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
ethernet5-3605/bridge	1	0	1	0	1	1	1	0	0	0	0	0	--	--

1 Bridge Interfaces displayed

Viewing bridge statistics-on-demand in bytes

View statistics on the Ethernet bridge interface in bytes.



Note: Bridge statistics are displayed in either packet counters or byte counters but not both.

1 Enable statistics on the specified Ethernet bridge interface to display in bytes.

```
zSH> bridge stats enable ethernet5-3605/bridge bytes
```

on-demand stats on interface "ethernet5-3605/bridge" have been enabled (bytes)

2 View statistics on the Ethernet bridge interface in bytes.

```
zSH> bridge stats ethernet5-3605/bridge
```

Interface Name	Received Packets				Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	Error	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
ethernet5-3605/bridge	--	--	--	--	--	--	--	--	0	0	0	0	400	332

1 Bridge Interfaces displayed

3 View statistics on the Ethernet bridge interface as rate-per-second for bytes.

zSH> **bridge rates ethernet5-3605/bridge**

Interface Name	Received Packets/Sec				Transmitted Packets/Sec				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	Error	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
ethernet5-3605/bridge	--	--	--	--	--	--	--	--	0	0	0	0	1	1

1 Bridge Interfaces displayed

Viewing statistics-on-demand per VLAN ID

1 Enter the **bridge stats vlanid** command to display bridge statistics per VLAN ID. Bridge statistics are displayed in default packet counters.

zSH> **bridge stats vlan 3605**

Interface Name	Received Packets				Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	Error	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
ethernet5-3605/bridge	14	0	2	--	14	119	611	0	0	0	0	0	--	--
1-b-5-0-eth-3605/bridge	--	--	--	--	--	--	--	--	0	0	0	0	--	--

2 Bridge Interfaces displayed

In this case, bridge statistics are displayed on the enabled *ethernet5-3604/bridge* interface and are not displayed on the *1-b-5-0-3605/bridge* interface as statistics-on-demand were not enabled for this interface.

2 Enable statistics on the Ethernet bridge interface to display in bytes per VLAN ID.

zSH> **bridge stats enable ethernet5-3605/bridge bytes**

on-demand stats on interface "ethernet5-3605/bridge" have been enabled (bytes)

3 View bridge statistics per VLAN ID in bytes.

zSH> **bridge stats vlan 3605**

Interface Name	Received Packets				Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	Error	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
ethernet5-3605/bridge	--	--	--	--	--	--	--	--	0	0	0	0	4972	3300
1-b-5-0-eth-3605/bridge	--	--	--	--	--	--	--	--	0	0	0	0	--	--

2 Bridge Interfaces displayed

View bridge statistics on GPON bridges

Bridge statistics are enabled and displayed on the GEM port for GPON bridges.

Viewing statistics on GPON bridge egress

Enter the **bridge stats interfaceName/bridge** command. Statistics on GPON egress are enabled by default.

zSH> **bridge stats 1-6-1-301-gponport-998/bridge**

Interface Name	Received Packets				Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	Error	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
1-6-1-301-gponport-998/bridge	--	--	--	--	0	5141	2311	0	0	0	0	0	--	--

1 Bridge Interfaces displayed

Enabling statistics-on-demand on GPON bridge for ingress

The **bridge stats enable** command enables the display of received packet information in the **bridge stats** command.

- 1 Enter the **bridge stats enable interfaceName/bridge** command.

```
zSH> bridge stats enable 1-6-1-401-gponport-3101/bridge
on-demand stats on interface "1-6-1-401-gponport-3101/bridge" have been enabled
```

- 2 Enter the **bridge stats** command to view the statistics.

```
zSH> bridge stats 1-6-1-401-gponport-3101/bridge
```

Interface Name	Received Packets			Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	Bcast	Error	UCast	MCast	Bcast	Alarm	Received	Transmitted
1-6-1-401-gponport-3101/bridge	279	1987	596	212	0	7985	0	0	0	0	0	--	--

1 Bridge Interfaces displayed

The statistics displayed are for the GEM port 401.

- 3 Verify the interface(s) is enabled for statistics-on-demand.

```
zSH> bridge stats list
1-6-1-401-gponport-3101/bridge
1 bridges have on-demand stats enabled
```

Viewing bridge statistics with rates

Enter the bridge rates command to view the rate counters as packets-per-seconds.

```
zSH> bridge rates 1-6-1-401-gponport-3101/bridge
```

Interface Name	Received Packets/Sec			Transmitted Packets/Sec				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	Bcast	Error	UCast	MCast	Bcast	Alarm	Received	Transmitted
1-6-1-401-gponport-3101/bridge	1	1	1	1	0	1	0	0	0	0	0	--	--

1 Bridge Interfaces displayed

Enabling statistics-on-demand on GPON bridge for ingress in byte counters

- 1 Enter the **bridge stats enable interfaceName/bridge bytes** command to view statistics with byte counters.

```
zSH> bridge stats enable 1-6-1-401-gponport-3101/bridge bytes
on-demand stats on interface "1-6-1-401-gponport-3101/bridge" have been enabled (bytes)
```

- 2 View the statistics in bytes.

```
zSH> bridge stats 1-6-1-401-gponport-3101/bridge
```

Interface Name	Received Packets			Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	Bcast	Error	UCast	MCast	Bcast	Alarm	Received	Transmitted
1-6-1-401-gponport-3101/bridge	--	--	--	--	--	--	--	0	0	0	0	9979	14077

1 Bridge Interfaces displayed

Viewing bridge byte counters in rate-per-second

Enter the **bridge rates** command to view bridge statistics in rate-per-second. In this case, the bridge interface is configured to view byte counters.

```
zSH> bridge rates 1-6-1-401-gponport-3101/bridge
```

Interface Name	Received Packets/Sec			Transmitted Packets/Sec				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
1-6-1-401-gponport-3101/bridge	--	--	--	--	--	--	--	0	0	0	0	4	5

1 Bridge Interfaces displayed

Viewing bridge packet counters in rate-per-second

If the bridge interfaces is configured for bytes, enter the **bridge statistics enable interfaceName/bridge** command to view bridge statistics in packets.

1 Enable bridge statistics for packets.

```
zSH> bridge stats enable 1-6-1-401-gponport-3101/bridge
```

on-demand stats on interface "1-6-1-401-gponport-3101/bridge" have been enabled

2 View the packets in rate-per-second.

```
zSH> bridge rates 1-6-1-401-gponport-3101/bridge
```

Interface Name	Received Packets/Sec			Transmitted Packets/Sec				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
1-6-1-401-gponport-3101/bridge	1	1	1	1	0	1	0	0	0	0	0	--	--

1 Bridge Interfaces displayed

Viewing GPON statistics-on-demand on the physical interface

Entering the physical interface at the ONU level displays all bridge interfaces.

Enter the **bridge stats** command on the physical interface.

```
zSH> bridge stats 1/6/1/1/gpononu
```

Interface Name	Received Packets			Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
1-6-1-301-gponport-998/bridge	0	0	0	0	2043	7840	0	0	0	0	0	--	--
1-6-1-401-gponport-3101/bridge	287	2045	613	218	0	8213	0	0	0	0	0	--	--
1-6-1-610-gponport-1001/bridge	0	0	0	0	0	0	0	0	0	0	0	--	--

3 Bridge Interfaces displayed

Viewing statistics on all bridge interfaces

1 Enter the bridge stats command to view all bridge statistics.

```
zSH> bridge stats
```

Interface Name	Received Packets			Transmitted Packets				Storm Detect Packets				Byte Counters	
	UCast	MCast	BCast	UCast	MCast	BCast	Error	UCast	MCast	BCast	Alarm	Received	Transmitted
ethernet2-3101/bridge	--	--	--	--	--	--	--	0	0	0	0	3590k	291k
ethernet2-998/bridge	241	6	144	0	0	0	0	0	0	0	0	--	--
ipobridge-3150/bridge	2440	4	244	213	0	0	0	0	0	0	0	--	--
ethernet4/bridge	--	--	--	--	--	--	--	0	0	0	0	--	--
ethernet2-3150/bridge	--	--	--	--	--	--	--	0	0	0	0	--	--
1-6-1-302-gponport-998/bridge	--	--	--	0	0	0	1914	0	0	0	0	--	--
1-6-1-402-gponport-3101/bridge	--	--	--	0	0	0	18530	0	0	0	0	--	--

```

1-6-1-301-gponport-998/bridge  --  --  --  0  83  29  0  0  0  0  0  0  0  --  --
1-6-1-401-gponport-3101/bridge 172 86 16 248 0 503 0 0 0 0 0 0 0 0  --  --
9 Bridge Interfaces displayed

```

2 Enter the bridge rates command to view bridge statistics in rate-per-second.

zSH> **bridge rates**

```

Interface                                     Received Packets/Sec  Transmitted Packets/Sec  Storm Detect Packets  Byte Counters
Name                                         UCast  MCast  BCast  UCast  MCast  BCast  Error  UCast  MCast  BCast  Alarm  Received  Transmitted
ethernet2-3101/bridge                       --  --  --  --  --  --  --  0  0  0  0  41  4
ethernet2-998/bridge                        1  1  1  0  0  0  0  0  0  0  0  --  --
ipobridge-3150/bridge                       1  1  1  1  0  0  0  0  0  0  0  --  --
ethernet4/bridge                           --  --  --  --  --  --  --  0  0  0  0  --  --
ethernet2-3150/bridge                       --  --  --  --  --  --  --  0  0  0  0  --  --
1-6-1-302-gponport-998/bridge               --  --  --  0  0  0  1  0  0  0  0  --  --
1-6-1-402-gponport-3101/bridge              --  --  --  0  0  0  1  0  0  0  0  --  --
1-6-1-301-gponport-998/bridge               --  --  --  0  1  1  0  0  0  0  0  --  --
1-6-1-401-gponport-3101/bridge              1  1  1  1  0  1  0  0  0  0  0  --  --
9 Bridge Interfaces displayed

```

3 Enter the bridge stats interfaceName/bridge command to view statistics on IP on bridge interfaces. Statistics are enabled by default.

zSH> **bridge stats ipobridge-3150/bridge**

```

Interface                                     Received Packets  Transmitted Packets  Storm Detect Packets  Byte Counters
Name                                         UCast  MCast  BCast  UCast  MCast  BCast  Error  UCast  MCast  BCast  Alarm  Received  Transmitted
ipobridge-3150/bridge                       2630  4  245  214  0  0  0  0  0  0  0  0  --  --
1 Bridge Interfaces displayed

```

4 Enter the bridge rates interfaceName/bridge command to view statistics on IP on bridge interfaces in rates-per-second.

zSH> **bridge rates ipobridge-3150/bridge**

```

Interface                                     Received Packets/Sec  Transmitted Packets/Sec  Storm Detect Packets  Byte Counters
Name                                         UCast  MCast  BCast  UCast  MCast  BCast  Error  UCast  MCast  BCast  Alarm  Received  Transmitted
ipobridge-3150/bridge                       1  1  1  1  0  0  0  0  0  0  0  0  --  --
1 Bridge Interfaces displayed

```

5 Enter the bridge show command to view all bridge interfaces.

zSH> **bridge show**

```

Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
dwn-vid  Tagged 998 1/6/1/1/gpononu 1-6-1-301-gponport-998/bridge UP
dwn  Tagged 1001 1/6/1/1/gpononu 1-6-1-610-gponport-1001/bridge UP
dwn  Tagged 3101 1/6/1/1/gpononu 1-6-1-401-gponport-3101/bridge UP D 00:13:72:dd:4b:67
D 10.51.1.200
dwn-vid  Tagged 998 1/6/1/2/gpononu 1-6-1-302-gponport-998/bridge DWN
dwn  Tagged 3101 1/6/1/2/gpononu 1-6-1-402-gponport-3101/bridge DWN
upl  Tagged 998 1/a/2/0/eth ethernet2-998/bridge UP S VLAN 998 default
upl  Tagged 3101 1/a/2/0/eth ethernet2-3101/bridge UP S VLAN 3101 default
tls  Tagged 3150 1/a/2/0/eth ethernet2-3150/bridge UP D f8:66:f2:0d:3c:41
D c4:7d:4f:a3:04:b4
dwn-vid 998 1/a/4/0/eth ethernet4/bridge DWN
ipobtls Tagged 3150 1/a/6/0/ipobridge ipobridge-3150/bridge UP S 00:01:47:93:74:54
S 10.51.50.118

```

10 Bridge Interfaces displayed

Bridge statistics display

Table 36 defines the columns the **bridge stats** and **bridge stats rules** commands display.

Table 36: bridge stats display columns

Column	Description
enabled	The on-demand stats collection for this bridge interface will be enabled and packets will be counted.
enabled, bytes	The on-demand stats collection for this bridge interface will be enabled and bytes will be counted.
ucastRx	Unicast packets received.
mcastRx	Multicast packets received.
bcastRx	Broadcast packets received.
ucastTx	Unicast packets sent.
mcastTx	Multicast packets sent.
errorTx	Error packets sent.
RulesSupported	The number of supported ingress statistics available for a line card.
RulesRemaining	The number of remaining ingress statistics available for a line card.
UcastPktBlocked	The number of unicast packets dropped due to bridge packet storm detection threshold exceeded.
McastPktBlocked	Number of multicast packets dropped due to bridge packet storm detection threshold exceeded.
BcastPktBlocked	Number of broadcast packets dropped due to bridge packet storm detection threshold exceeded.
AlarmCnt	This counter reflects the number of times this interface has transitioned to the alarm state due to the bridge packet storm detection threshold being exceeded for a pre-defined number of seconds.
bytesRcvd	This is a count of the number of bytes received. On-demand stats must be enabled for byte counters otherwise this counter is zero.
bytesSent	This is a count of the number of bytes transmitted. On-demand stats must be enabled for byte counters otherwise this counter is zero.

Administrative commands

This section describes some of the most useful bridge commands:

- [bridge add/delete commands, page 424](#)
- [bridge show/showall commands, page 424](#)
- [bridge-path add/modify/show/delete commands, page 425](#)



Note: Entering general CLI commands on systems with large GPON configurations can take a long time to process. You must be as specific as possible with CLI commands. For example, **bridge flush all** should not be used. Instead, use commands based on the specific interface or MAC address.

bridge add/delete commands

Add a bridge interface on the specified physical interface with the **bridge add interface/type** command.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 0 slan 501 tagged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-0-501/bridge
Bridge-path added successfully
```

The **bridge delete** command deletes a specific bridge entry from the system. Delete a bridge interface from the specified physical interface(s). Tagging/vlan/slan act as qualifying parameters. If 'all' is specified, all bridges found matching the qualifiers are deleted. If 'all' is not specified, you must enter sufficient qualifiers to make identification of target bridge unambiguous.

```
zSH> bridge delete ethernet5-0-501/bridge vlan 0 slan 501
Bridge-path deleted successfully
ethernet5-0-501/bridge delete complete
```

bridge show/showall commands

The **bridge show** and **bridge showall** commands display either a single bridge path entry or the entire bridge table.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
dwn-dat      Tagged  200  1/6/1/0/eth      1-6-1-0-eth-200/bridge  UP
upl          Tagged  100  1/a/5/0/eth      ethernet5-100/bridge    UP
upl          Tagged  200  1/a/5/0/eth      ethernet5-200/bridge    UP  S VLAN 200 default
upl          Tagged  600  1/a/5/0/eth      ethernet5-600/bridge    UP  S VLAN 600 default
ipobdwn      Tagged  600  1/a/6/0/ipobridge ipobridge-600/bridge    UP  S 00:01:47:11:b7:c6
                                           S 192.168.8.21

5 Bridge Interfaces displayed
```

```
zSH> bridge showall
```

```

      Orig
Type   VLAN/SLAN  VLAN/SLAN  Physical          Bridge          St  Table Data
-----
dwn-dat      Tagged  200    1/6/1/0/eth      1-6-1-0-eth-200/bridge  UP
upl          Tagged  200    1/a/5/0/eth      ethernet5-200/bridge    UP  S VLAN 200 default
upl          Tagged  600    1/a/5/0/eth      ethernet5-600/bridge    UP  S VLAN 600 default
ipobdwn      Tagged  600    1/a/6/0/ipobridge ipobridge-600/bridge    UP  S 00:01:47:11:b7:c6
                                           I=1744 A=4 U=277 F=0
                                           S 192.168.8.21
                                           I=1744 A=277 U=277 F=0

```

```
4 Bridge Interfaces displayed
```

```
Aging counter: 4931
```

```
Renew failed: 0
```

```
Filter renewed: 0
```

```
Flap Suppresses: 0
```

bridge-path add/modify/show/delete commands

Most bridge-paths are automatically created with the **bridge add** command and VLAN ID. The bridge-path is a static-bridge assignment between an existing VLAN/address and an interface.

The **bridge-path add** command is used when configuring secure bridging. See [Static IP and MAC for secure bridging on the MXK, page 284](#)

```
zSH> bridge add 1-a-5-0/eth uplink vlan 100
```

```
Adding bridge on 1-a-5-0/eth
```

```
Created bridge-interface-record ethernet5-100/bridge
```

```
Bridge-path added successfully
```

Modify a bridge-path to enable certain features, in this case configuring loop prevention with **blockAsym**.

```
zSH> bridge-path modify ethernet5-100/bridge vlan 100 default block blockAsym
```

```
Bridge-path ethernet5-100/bridge/3/100/0/0/0/0/0/0/0 has been modified
```

The **bridge-path show** command lists all static-bridge assignments.

```
zSH> bridge-path show
```

```

VLAN/SLAN  Bridge          Address
-----
      200 ethernet5-200/bridge  Default, Age: 3600, MCAST Age: 250, IGMP
Query Interval: 0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym
      600 ethernet5-600/bridge  Default, Age: 3600, MCAST Age: 250, IGMP
Query Interval: 0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym
      600 ipobridge-600/bridge  192.168.8.21
      600 ipobridge-600/bridge  00:01:47:11:b7:c6
      100 ethernet5-100/bridge  Default, Age: 3600, MCAST Age: 250, IGMP
Query Interval: 0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym

```

The **bridge-path delete** command deletes a static-bridge assignment and breaks an association between a VLAN/address and an interface.

```
zSH> bridge-path delete ethernet5-100/bridge vlan 100 default
```

Delete complete

5

VIDEO CONFIGURATION

This chapter explains how to configure the MXK for bridged video and includes:

- [MXK bridged video overview, page 427](#)
- [MXK bridged video with IGMP proxy, page 428](#)
- [MXK basic bridged video configuration, page 429](#)
- [Advanced bridged video with IGMP and IGMP DSCP configuration, page 433](#)
- [Advanced bridged video on the MXK with VLAN translation and MVR, page 439](#)
- [Display bridge IGMP, page 462](#)

MXK bridged video overview

Video bridging enables video packets to be forwarded over bridges from a headend device down to downstream device. In this case, the video travels from the source, or head-end device, using one video stream to passively traverse the MXK backplane. This lowers the bandwidth requirements for video packets traversing the MXK.

Video bridging requires configuring an uplink bridge and a downlink-video bridge. The uplink bridge is associated with a location that contains the video content that allows the MXK to receive video streams from the network. The bridge interface only transmits multicast traffic for which a JOIN request is received.

The downlink-video bridge is associated with interfaces that have hosts connected to them and allows the MXK to send video groups from downlink interfaces to the network.

Depending on the transit type, most *downlink-video* bridges are *tagged*, except for ADSL.

Note that JOIN requests enter on a interface associated with a downlink bridge and pass through on a interface associated with an uplink bridge.

MXK bridged video with IGMP proxy

This section describes IGMP proxy and join and leave requests:

- [IGMP proxy overview, page 428](#)
- [IGMP proxy join and leave requests, page 428](#)

IGMP proxy overview

Enabling IGMP proxy reduces traffic between the MXK and the upstream multicast headend device by changing the behavior of the MXK for more efficient tracking and grouping of JOIN and LEAVE requests. MXK IGMP proxy also supports the following:

- Solicited or unsolicited query reports.
- Queries are sent only to hosts that have sent a join request.
- Compliance with rfc4541 regarding IGM forwarding and data rules.
- Information table is available during redundant uplink port switchovers.
- Membership reports on downlink bridges are not forwarded.
- When join requests are received without a leave, it is assumed that the set top box is watching both channels.
- MXK IGMP proxy supports existing Max Video Streams and Multicast Control List functionality.
- Using the IP on a bridge IP address when a join request is sent to the upstream multicast headend device.

IGMP proxy join and leave requests

For video without IGMP proxy, join requests from downstream hosts are simply forwarded by the MXK to the multicast headend device. With IGMP proxy, join requests from downstream hosts are not forwarded by the MXK to the multicast headend device in the network, but are tracked by the MXK in an information table where hosts are organized into a group. When a host sends a join request that is the first join request of the group, the MXK terminates the join request from the host, originates a new join request, and sends it to the multicast headend device in the network with the default IP address of 10.10.10.1 and a MAC address.

When a host sends a leave request that is the last leave request of the group, the MXK terminates the leave request from the host and originates a new leave request and sends it to the multicast headend device in the network. All leave requests, regardless of whether they are the last leave request of the group, or any earlier leave requests, are terminated on the MXK.

In this way, the multicast headend device starts and stops video transmission by processing requests sent directly from the MXK and not from downstream

hosts. IGMP proxy is when the MXK sends join and leave requests to the network and monitors the join and leave requests from hosts to the MXK.

MXK basic bridged video configuration

This section describes how to configure the MXK for video connections so that traffic passes between the MXK, the upstream video source, and the subscriber:

Basic bridged video with IGMP proxy configuration overview

Bridged video connections require bridge configurations on the uplink and on the downlink.

Generally, these are the steps to follow to configure the MXK for bridged video.

Configuring a basic video connection on the MXK

- 1 Create an uplink bridge on a FE/GE uplink port with VLAN ID and IGMP proxy.

See [Creating an uplink bridge on an Ethernet uplink port for video on page 429](#).

- 2 Create the multicast control lists, if necessary.

See [Creating multicast control lists on page 430](#).

- 3 Create a downlink-video bridge with a VLAN ID and specify the maximum number of video streams and a multicast control list.

See [Creating a downlink bridge on a Ethernet port for video services on page 431](#).

Basic video configuration with IGMP proxy

You must create an uplink bridge on a FE/GE uplink and configure the bridge for video service with VLAN ID and IGMP proxy and then create a downlink bridge to the subscriber.

Creating an uplink bridge on an Ethernet uplink port for video

You create a video bridge on the uplink by first creating an uplink bridge on an Ethernet port with the **bridge add** command using a VLAN ID. Then enter the multicast aging period and IGMP query interval for video traffic when entering the **bridge-path add** command.

- 1 Create a tagged uplink bridge with a VLAN ID and the keyword **igmpproxy**. Designating **igmpproxy** enables IGMP proxy.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 101 igmpproxy
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-101/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
upl          Tagged 101  1/a/5/0/eth  ethernet5-101/bridge  UP  S VLAN 101 default
1 Bridge Interfaces displayed
```

2 View the bridge path for the bridge interface with IGMP proxy enabled.

```
zSH> bridge-path show
VLAN/SLAN  Bridge  Address
-----
101 ethernet5-101/bridge  Default, Age: 3600, MCAST Age: 250,
IGMP Query Interval: 120, IGMP Proxy, IGMP DSCP: 0, Flap Mode: Default, Block: Asym
```

Creating multicast control lists

Specifying a multicast control list of 0 allows all IP multicasts.

The downlink bridge is configured for video by entering the keyword **video** and the multicast control list and maximum number of video streams in the *m/n* format with the new mcast-control-entry command.

new mcast-control-entry <m>/<n>

<m> is the multicast-control-list ID number and <n> is an entry index to the multicast-control-list <m>

The new multicast-control-list <m>/<n>, where <m> is the multicast-control-list ID number, and <n> is an entry index to the multicast-control-list <m>.

Each multicast-control-list <m> usually has several entry records <n>.

1 The following example adds three entries to multicast list 1:

```
zSH> new mcast-control-entry 1/1
mcast-control-entry 1/1
Please provide the following: [q]uit.
ip-address: -> {0.0.0.0}: 224.1.1.1
type: -----> {normal}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

zSH> new mcast-control-entry 1/2
mcast-control-entry 1/2
Please provide the following: [q]uit.
ip-address: -> {0.0.0.0}: 224.1.1.24
type: -----> {normal}:
```

```
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

```
zSH> new mcast-control-entry 1/3
mcast-control-entry 1/3
Please provide the following: [q]uit.
ip-address: -> {0.0.0.0}: 224.1.1.25
type: -----> {normal}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Continue adding as many multicast entries as necessary.

2 Verify the multicast entries:

```
zSH> mcast show mcl 1
MCAST CONTROL LIST : 1
                224.1.1.1          224.1.1.24 224.1.1.25
```

Creating a downlink bridge on a Ethernet port for video services

The syntax for the downlink bridge:

```
bridge add <interface/type> vc <vpi/vci> td <td val> downlink vlan
<vlanid> [untagged][tagged] video <mcastControlListID>/<maxMulticast>
```

Create a downlink bridge with VLAN ID on an ADSL port.

A multicast control list entry of 0 allows all IP multicasts.

```
zSH> bridge add 1-1-6-0/eth downlink-video vlan 101 tagged video 0/6
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth-101/bridge
```

Verify the bridge.

```
zSH> bridge show
```

Type	Orig	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-vid		Tagged	101	1/1/6/0/eth	1-1-6-0-eth-101/bridge	UP	
upl		Tagged	101	1/a/5/0/eth	ethernet5-101/bridge	UP	S VLAN 101 default

2 Bridge Interfaces displayed

Deleting the video configuration

If necessary, you can delete the uplink bridge, bridge path, multicast control lists, and downlink bridges.

1 Delete the multicast control lists.

```
zSH> delete mcast-control-entry 1/1
mcast-control-entry 1/1
1 entry found.
```

```
Delete mcast-control-entry 1/1? [y]es, [n]o, [q]uit : y  
mcast-control-entry 1/1 deleted.
```

```
zSH> delete mcast-control-entry 1/2  
mcast-control-entry 1/2  
1 entry found.  
Delete mcast-control-entry 1/2? [y]es, [n]o, [q]uit : y  
mcast-control-entry 1/2 deleted.
```

```
zSH> delete mcast-control-entry 1/3  
mcast-control-entry 1/3  
1 entry found.  
Delete mcast-control-entry 1/3? [y]es, [n]o, [q]uit : y  
mcast-control-entry 1/3 deleted.
```

2 Delete the downlink-video bridge.

```
zSH> bridge delete 1-1-6-0-eth-101/bridge vlan 101  
1-1-6-0-eth-101/bridge delete complete
```

3 Delete the uplink bridge.

```
zSH> bridge delete ethernet5-101/bridge vlan 101  
Bridge-path deleted successfully  
ethernet5-101/bridge delete complete
```

Advanced bridged video with IGMP and IGMP DSCP configuration

This section describes IGMP DSCP and includes:

- [IGMP DSCP overview, page 433](#)
- [IGMP DSCP and IGMP with proxy reporting and default IP address, page 435](#)
- [IGMP DSCP and IGMP with proxy reporting and custom IP address, page 436](#)

IGMP DSCP overview

The bridge-path can be used to specify the source IP and DSCP bits to use when sending IGMP packets to the network. The source IP is required by some routers to uniquely identify the origin of IGMP packets. The DSCP bits prioritize the IGMP packets through the edge/core network. See [Table 37](#) for DSCP core values.

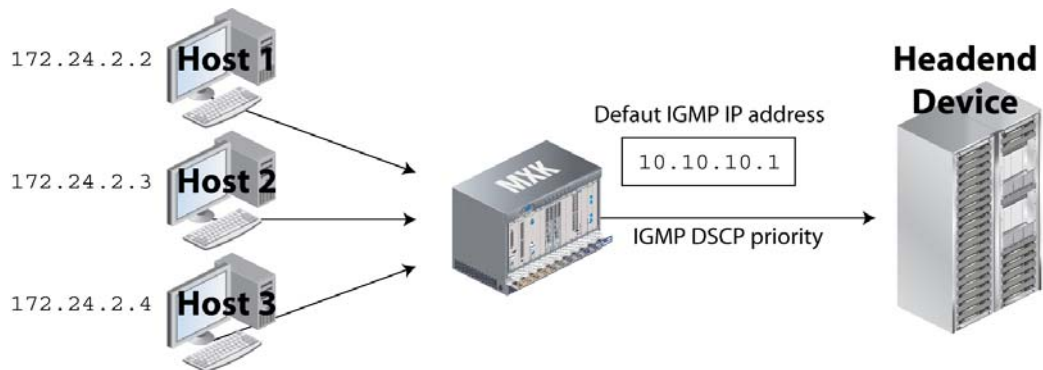
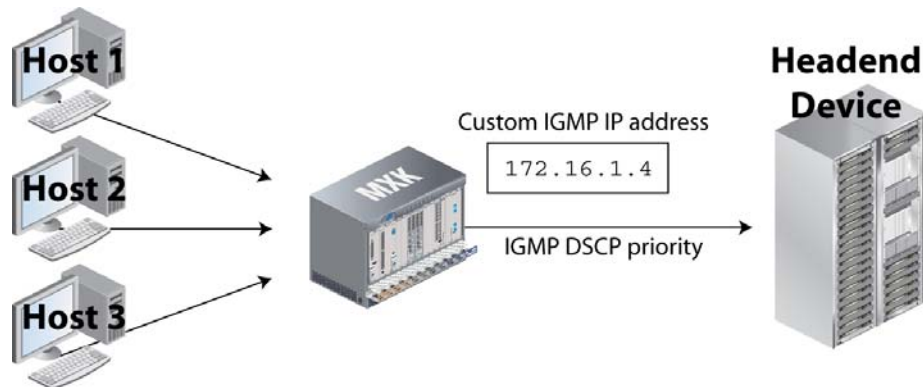
Table 37: DSCP code values

String	Value
af11	Mark packets with AF11 dscp (001010)
af12	Mark packets with AF12 dscp (001100)
af13	Mark packets with AF13 dscp (001110)
af21	Mark packets with AF21 dscp (010010)
af22	Mark packets with AF22 dscp (010100)
af23	Mark packets with AF23 dscp (010110)
af31	Mark packets with AF31 dscp (011010)
af32	Mark packets with AF32 dscp (011100)
af33	Mark packets with AF33 dscp (011110)
af41	Mark packets with AF41 dscp (100010)
af42	Mark packets with AF42 dscp (100100)
af43	Mark packets with AF43 dscp (100110)
cs1	Mark packets with CS1(precedence 1) dscp (001000)
cs2	Mark packets with CS2(precedence 2) dscp (010000)
cs3	Mark packets with CS3(precedence 3) dscp (011000)
cs4	Mark packets with CS4(precedence 4) dscp (100000)

Table 37: DSCP code values (Continued)

String	Value
cs5	Mark packets with CS5(precedence 5) dscp (101000)
cs6	Mark packets with CS6(precedence 6) dscp (110000)
cs7	Mark packets with CS7(precedence 7) dscp (111000)
default	Mark packets with default dscp (000000)
ef	Mark packets with EF dscp (101110)

When IGMP proxy is enabled on a static uplink bridge, the default source IP address in the Ethernet packet sent from the bridge is 10.10.10.0 as shown in [Figure 55](#). In certain cases there may be a need to replace 10.10.10.1 with a custom Ethernet IP address. For example when a router in the network has implemented Reverse Path Forwarding and expects an IP address in the subnet of the router or when different IP addresses in the same subnet are inserted for different SLMS devices for the purposes of debugging, see [Figure 56](#).

Figure 55: MXK with default IGMP IP address and IGMP DSCP priority**Figure 56: MXK with custom IGMP IP address and DSCP priority**

IGMP DSCP and IGMP with proxy reporting and default IP address

After creating the uplink bridge and enabling IGMP proxy to pass video traffic, use the **bridge-path modify** command to configure DSCP priority in IP packets for JOIN and LEAVE requests to the network. Enabling IGMP proxy sends the default IP address 10.10.10.1.

Configuring IGMP with proxy reporting and IGMP DSCP

- 1 Create an *tagged* uplink bridge on a n Ethernet port, designate a VLAN ID, and enable proxy reporting.

```
zSH> bridge add 1-a-7-0/eth uplink vlan 1001 tagged igmpproxy
Adding bridge on 1-a-7-0/eth
Created bridge-interface-record ethernet7-1001/bridge
Bridge-path added successfully
```

The default for uplink bridges with VLAN IDs is tagged.

Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
upl           Tagged 1001  1/a/7/0/eth  ethernet7-1001/bridge  DWN S VLAN 1001 default
1 Bridge Interfaces displayed
```

The default bridge path is created with IGMP proxy.

```
zSH> bridge-path show
VLAN/SLAN  Bridge  Address
-----
1001 ethernet7-1001/bridge  Default, Age: 3600, MCAST Age: 250, IGMP Query
Interval: 120, IGMP Proxy, IGMP DSCP: 0, Flap Mode: Default, Block: Asym
```

- 2 Modify the bridge-path for IGMP DSCP priority.

The **igmpDSCP** sets the DSCP priority for IGMP messages to the network.

```
zSH> bridge-path modify ethernet7-1001/bridge vlan 1001 default igmpDSCP af12
Bridge-path ethernet7-1001/bridge/3/1001/0/0/0/0/0/0/0/0 has been modified
```

Verify the bridge path with IGMP DSCP.

```
zSH> bridge-path show
VLAN/SLAN  Bridge  Address
-----
ym ethernet7-1001/bridge  Default, Age: 3600, MCAST Age: 250, IGMP Query
Interval: 120, IGMP Proxy, IGMP DSCP: af12, Flap Mode: Default, Block: Asym
```

Creating a downlink-video bridge on an Active Ethernet port with video streams and multicast control list

You can create a downlink bridge on an Active Ethernet port and specify a maximum number of video streams. Add the multicast control list and designate the maximum video streams using the *m/n* format. The multicast control list is set first and the maximum video streams second.

Entering 0 for the multicast control list allows all IP multicasts.

Create a downlink bridge on an Active Ethernet interface for video.

```
zSH> bridge add 1-1-6-0/eth downlink-video vlan 1001 tagged video 0/3
```

Adding bridge on 1-1-6-0/eth

Created bridge-interface-record 1-1-6-0-eth-1001/bridge

Verify the bridge.

```
zSH> bridge show
```

Type	Orig	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-vid		Tagged	1001	1/1/6/0/eth	1-1-6-0-eth-1001/bridge	UP	
upl		Tagged	1001	1/a/7/0/eth	ethernet7-1001/bridge	UP S	VLAN 1001 default

2 Bridge Interfaces displayed

IGMP DSCP and IGMP with proxy reporting and custom IP address

After creating the uplink bridge and enabling IGMP proxy to pass video traffic, use the **bridge-path modify** command to configure DSCP priority in IP packets for JOIN and LEAVE requests to the network and the custom IP address. Enabling IGMP proxy will send the custom IP address.

Configuring IGMP with proxy reporting, custom IP address, and IGMP DSCP

You can configure the MXK to send a custom IP address used in proxy on the bridge path along with IGMP DSCP for IGMP priority to the network.

- 1 Create an *tagged* uplink bridge on a n Ethernet port, designate a VLAN ID, and enable proxy reporting.

```
zSH> bridge add 1-a-7-0/eth uplink vlan 1002 tagged igmpproxy
```

Adding bridge on 1-a-7-0/eth

Created bridge-interface-record ethernet7-1002/bridge

Bridge-path added successfully

Verify the bridge.

```
zSH> bridge show
```

Type	Orig	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
upl		Tagged	1002	1/a/7/0/eth	ethernet7-1002/bridge	DWN S	VLAN 1002 default

1 Bridge Interfaces displayed

The default bridge path is created with IGMP proxy.

```
zSH> bridge-path show
VLAN/SLAN   Bridge                               Address
-----
1002 ethernet7-1002/bridge           Default, Age: 3600, MCAST Age: 250, IGMP Query
Interval: 120, IGMP Proxy, IGMP DSCP: 0, Flap Mode: Delock: Asym
```

2 Modify the bridge-path for IGMP DSCP priority and custom IP address.

The **igmpDSCP** sets the DSCP priority for IGMP messages to the network.

The **igmpsendip enable <ipaddress>** sends a custom IP address.

```
zSH> bridge-path modify ethernet7-1002/bridge vlan 1002 default igmpsendip enable
172.16.1.3 igmpDSCP af13
Bridge-path ethernet7-1002/bridge/3/1002/0/0/0/0/0/0/0 has been modified
```

Verify the bridge path.

```
zSH> bridge-path show
VLAN/SLAN   Bridge                               Address
-----
de: Default, Block: Asym ethernet7-1002/bridge           Default, Age: 3600, MCAST Age:
241, IGMP Query Interval: 120, IGMP Proxy, Custom IP 172.16.1.3, IGMP DSCP: af13, Flap Mode:
Default, Block: Asym
```

To revert to sending the default IP address of 10.10.10.1, enter **igmpsendip disable**.

Creating a downlink bridge on an Active Ethernet port with video streams and multicast control list

You can create a downlink bridge on an Active Ethernet port and specify a maximum number of video streams. Add the multicast control list and designate the maximum video streams using the *m/n* format. The multicast control list is set first and the maximum video streams second.

Entering *0* for the multicast control list allows all IP multicasts.

Create a downlink bridge on an Active Ethernet interface for video.

```
zSH> bridge add 1-1-6-0/eth downlink-video vlan 1002 tagged video 0/3
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth-1002/bridge
```

Verify the bridge.

```
zSH> bridge show
Orig
Type   VLAN/SLAN  VLAN/SLAN  Physical  Bridge
St  Table Data
-----
-----
```

Video Configuration

```
dwn-vid      Tagged 1002 1/1/6/0/eth      1-1-6-0-eth-1002/bridge
UP
upl          Tagged 1002 1/a/7/0/eth      ethernet7-1002/bridge
DWN S VLAN 1002 default
2 Bridge Interfaces displayed
```

Advanced bridged video on the MXK with VLAN translation and MVR

This section describes how to configure the MXK for video connections in bridging configurations that need to utilize VLAN translation, Multicast VLAN Registration (MVR), or both VLAN translation and MVR.

- [Bridged video on the MXK with VLAN translation, page 440](#)
- [Bridged video on the MXK with MVR, page 443](#)
- [Bridged video on the MXK with VLAN translation and MVR, page 447](#)
- [Bridged video on the MXK with dual MVR, page 456](#)

MVR allows video subscribers to share one multicast VLAN in the network while remaining in their own unique subscriber VLAN. MVR can send packets received from the multicast headend device on one MVR VLAN to one or more than one subscriber VLAN IDs.

In cases where the CPE devices have preconfigured VLANs or SLANs, the MXK supports VLAN translation, that is, the ability to translate preconfigured VLANs on the subscriber side to VLANs currently assigned on the network side.

For SLAN promotion *and* VLAN translation bridging configurations on the MXK, the name of the *tagged* bridge interface will include the interface, the translated to VLAN ID, and the SLAN ID.



Note: This feature is valid on single-slot Ethernet cards and VDSL2 combo cards with splitter.

For example,

```
zSH> bridge add 1-6-1-0/eth downlink vlan 100 xlate-to 1002 slan 500 mvrvlan 2220
tagged video 1/3
```

```
Adding bridge on 1-6-1-0/eth
```

```
Created bridge-interface-record 1-6-1-0-eth-1002-500/bridge
```

```
zSH> bridge show
```

Orig							
Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
dwn	100/----	Tg 1002/500	1/6/1/0/eth	1-6-1-0-eth-1002-500/bridge	UP	D	00:01:47:31:dc:1a
upl		ST 0/500	1/a/8/0/eth	ethernet8-0-500/bridge	DWN	S	SLAN 500 VLAN 0 default
mvr		Tagged 2220	1/a/8/0/eth	ethernet8-2220/bridge	DWN	S	MVR vlan 2220

3 Bridge Interfaces displayed

This feature is only supported on the Active Ethernet single-slot card and the VDSL combo card.

In cases where devices upstream from the MXK expect SLAN IDs, SLAN IDs can be promoted from *tagged* downstream bridges to *tagged* upstream bridges.

The range for translated VLAN IDs is 1-4090 (some VLANs are reserved).

VLAN translation and VLAN translation and promotion is supported on Ethernet (single-slot only), VDSL2 combo cards with splitter.

Possible bridging configuration behaviors for VLAN/SLAN for video configurations are:

- either the network facing or the subscriber facing bridge is *untagged*
VLAN translation not allowed
- subscriber facing single-tagged bridge, network facing single-tagged bridge with VLAN translation for video (*tagged to tagged*)
Refer to [Bridged video on the MXK with VLAN translation on page 440](#).
- subscriber facing single-tagged bridge, network facing single-tagged bridge for MVR (*tagged to tagged*)
Refer to [Bridged video on the MXK with MVR on page 443](#).
- subscriber facing single-tagged bridge, network facing single-tagged bridge with VLAN translation and MVR (*tagged to tagged*)
Refer to [Bridged video on the MXK with VLAN translation and MVR on page 447](#).
- subscriber facing single-tagged bridge to network facing double-tagged bridge with SLAN promotion and MVR (*tagged to tagged*)
Refer to [Bridged video on the MXK with SLAN promotion and MVR on page 450](#).
- subscriber facing single-tagged bridge with VLAN translation, SLAN promotion, and MVR (*tagged to tagged*)
Refer to [Bridged video on the MXK with VLAN translation, SLAN promotion, and MVR on page 453](#).

Bridged video on the MXK with VLAN translation

This section describes configuring asymmetric bridges on the MXK for basic VLAN translation and video.

When configuring the asymmetric bridges for basic VLAN translation, both the uplink and the downlink bridges are configured as *tagged*.

Any downlink or subscriber facing bridges configured for video must be *tagged*.

As shown in [Figure 57](#), the VLAN ID 200 on the downlink bridge is translated on the MXK to VLAN ID 1001 for the network facing uplink bridge.

IGMP proxy reporting, a feature of bridged video, sends the default IP address 10.10.10.0 to the multicast headend device.

For bridged video, IGMP proxy is enabled in two ways.

- When an uplink bridge is configured for video without an MVR VLAN, the keyword **igmpproxy** is entered with the **bridge add** command and IGMP proxy is enabled.
- When the uplink bridge is configured for video with an MVR VLAN, the keyword **mvr <vlan id>** is entered with the **bridge add** command and IGMP proxy is enabled.

Figure 57: Asymmetric bridging with VLAN translation and video



Creating single-tagged to single-tagged asymmetric bridged video for VLAN translation

- 1 Create a *tagged* uplink bridge with VLAN ID on a Ethernet port on the uplink card.

```
zSH> bridge add 1-a-7-0/eth uplink vlan 1001 tagged igmpproxy
Adding bridge on 1-a-7-0/eth
Created bridge-interface-record ethernet7-1001/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
      Orig
Type VLAN/SLAN  VLAN/SLAN      Physical      Bridge
St  Table Data
-----
upl           Tagged 1001    1/a/7/0/eth      ethernet7-1001/bridge
DWN S VLAN 1001 default
1 Bridge Interfaces displayed
```

Verify the bridge path. The IGMP Proxy is displayed indicating IGMP proxy is enabled.

```
zSH> bridge-path show
VLAN/SLAN  Bridge      Address
-----
      1001 ethernet7-1001/bridge      Default, Age: 3600, MCAST Age: 250,
IGMP Query Interval: 120, IGMP Proxy, IGMP DSCP: 0, Flap Mode: Default, Block: Asym
```

- 2 Add the bridge path for the uplink bridges to pass video traffic by setting the multicast aging period and the IGMP query interval.

Although default bridge paths are created with the **bridge add** command, they can be created again with the both the default configuration information and the multicast and IGMP settings.

The **mcast** sets the maximum age, in seconds, of a multicast packet before it is purged.

The **igmptimer** indicates a time value in seconds. This value should be greater than 0. If you enter 0, the querying function is disabled.

```
zSH> bridge-path modify ethernet7-1001/bridge vlan 1001 default mcast 90 igmptimer 30
Bridge-path ethernet7-1001/bridge/3/1001/0/0/0/0/0/0/0 has been modified
```

Verify the bridge paths.

```
zSH> bridge-path show
VLAN/SLAN   Bridge                                     Address
-----
1001 ethernet7-1001/bridge               Default, Age: 3600, MCAST Age: 90,
IGMP Query Interval: 30, IGMP Proxy, IGMP DSCP: 0, Flap Mode: Default, Block: Asym
```



Note: If your network checks for source IP addresses, the default proxy IP address can be configured to a custom IP address.

igmpsendip is set to *enable* with the custom IP address.

Configure the bridge path with a custom IP address for proxy.

```
zSH> bridge-path modify ethernet7-1001/bridge vlan 1001 default mcast 90 igmptimer 30 igmpsendip enable 172.16.24.1
Bridge-path ethernet7-1001/bridge/3/1001/0/0/0/0/0/0/0 has been modified
```

Verify the bridge path.

```
zSH> bridge-path show
VLAN/SLAN   Bridge                                     Address
-----
1001 ethernet7-1001/bridge               Default, Age: 3600, MCAST Age: 90,
IGMP Query Interval: 30, IGMP Proxy, Custom IP 172.16.24.1, IGMP DSCP: 0, Flap
Mode: Default, Block: Asym
```

3 Create the downlink bridge for VLAN translation and video.

The *tagged* downlink bridge is configured with the subscriber facing VLAN ID and the **xlate-to** VLAN ID for the uplink bridge.

Add the multicast control list and designate the maximum video streams using the *m/n* format. The multicast control list is set first and the maximum video streams second.

Members of the multicast control list must be defined to receive the video signal and is entered first in the *m/n* format.

Entering 0 for the multicast control list allows all IP multicasts.

```
zSH> bridge add 1-1-6-0/eth downlink-video vlan 200 xlate-to 1001 tagged video 0/2
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth-1001/bridge
```

- 4 Verify the bridges. The **bridge show** command displays the VLAN ID of the downlink bridge(s) and the VLAN ID the MXK translated.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
dwn-vid  200      Tagged 1001  1/1/6/0/eth  1-1-6-0-eth-1001/bridge  UP  D 00:02:71:2e:2b:61
upl      Tagged 1001  1/a/7/0/eth  ethernet7-1001/bridge    DWN S VLAN 1001 default
2 Bridge Interfaces displayed
```

Deleting single-tagged to single-tagged bridged video with VLAN translation

- 1 Verify the bridges.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
dwn-vid  200      Tagged 1001  1/1/6/0/eth  1-1-6-0-eth-1001/bridge  UP  D 00:02:71:2e:2b:61
upl      Tagged 1001  1/a/7/0/eth  ethernet7-1001/bridge    DWN S VLAN 1001 default
2 Bridge Interfaces displayed
```

- 2 Delete the uplink bridges.



Note: The **bridge delete** command automatically deletes the uplink bridge path.

```
zSH> bridge delete ethernet7-1001/bridge vlan 1001
Bridge-path deleted successfully
ethernet7-1001/bridge delete complete
```

- 3 Delete the downlink bridge. Bridges with VLAN ID translation use the translated VLAN ID in the **bridge delete** syntax.

```
zSH> bridge delete 1-6-1-0-eth-1001/bridge vlan 1001
1-6-1-0-eth-1001/bridge delete complete
```

Bridged video on the MXK with MVR

This section describes configuring asymmetric bridges on the MXK with MVR for IGMP and video.

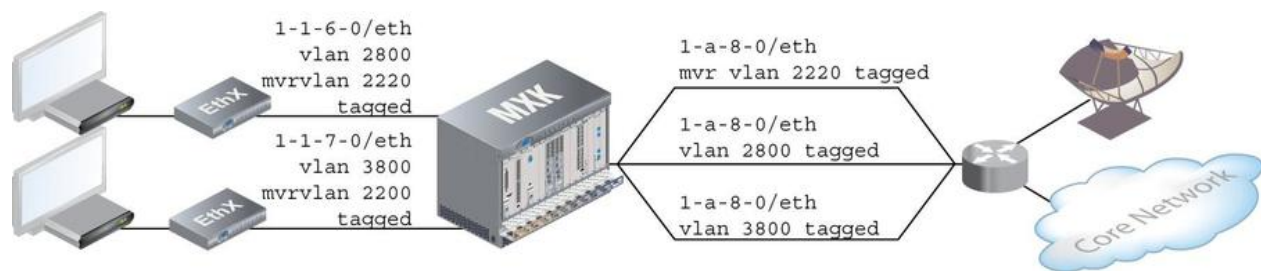
When configuring a bridge for MVR video, you create an MVR bridge for the downstream multicast video, and uplink bridges for everything that is not downstream multicast.

MVR bridges are always *tagged*. Any bridge that passes multicast traffic *must* be *tagged*.

In this configuration, the uplink bridge, the MVR bridge, and the downlink bridge are *tagged*.

As shown in [Figure 58](#), the MVR bridge with MVR VLAN ID can be used by multiple downlink bridges for downstream multicast video.

Figure 58: Asymmetric bridges with MVR and video



Creating single-tagged to single-tagged asymmetric bridged video with MVR

This case describes how one bridge configured with the MVR VLAN is used by multiple downstream bridges.

- 1 Create a *tagged* MVR bridge with VLAN ID on an uplink Ethernet port for all downstream multicast traffic.

```
zSH> bridge add 1-a-8-0/eth mvr vlan 2220 tagged
Adding bridge on 1-a-8-0/eth
Created bridge-interface-record ethernet8-2220/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
mvr          Tagged 2220  1/a/8/0/eth  ethernet8-2220/bridge  DWN S MVR vlan 2220
1 Bridge Interfaces displayed
```

Verify the bridge path.

```
zSH> bridge-path show
VLAN/SLAN  Bridge  Address
-----
2220 ethernet8-2220/bridge  MVR, MCAST Age: 250, IGMP Query Interval: 120,
IGMP Proxy, IGMP DSCP: 0
```

The video defaults created on MVR bridge paths are:

- IGMP proxy reporting is enabled and sends the default IP address 10.10.10.0
- **mcast** is set to 250 seconds
- **igmp timer** is set to 120 seconds



Note: If your network checks for the source IP addresses, the default proxy IP address can be configured to a custom IP address.

igmpsendip is set to *enable* with the custom IP address.

Configure the bridge path with a custom IP address for proxy.

```
zSH> bridge-path modify ethernet8-2220/bridge vlan 2220 mvr igmpsendip enable
172.16.24.1
Bridge-path ethernet8-2220/bridge/13/2220/0/0/0/0/0/0/0 has been modified
```

Verify the bridge path.

```
zSH> bridge-path show
VLAN/SLAN    Bridge                                     Address
-----
2220 ethernet8-2220/bridge                MVR, MCAST Age: 241, IGMP Query Interval: 120,
IGMP Proxy, Custom IP 172.16.24.1, IGMP DSCP: 0
```

2 Create *tagged* uplink bridges for all traffic except downstream multicast traffic.

```
zSH> bridge add 1-a-8-0/eth uplink vlan 2800 tagged
Adding bridge on 1-a-8-0/eth
Created bridge-interface-record ethernet8-2800/bridge
Bridge-path added successfully
```

```
zSH> bridge add 1-a-8-0/eth uplink vlan 3800 tagged
Adding bridge on 1-a-8-0/eth
Created bridge-interface-record ethernet8-3800/bridge
Bridge-path added successfully
```

Verify the bridges.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical      Bridge          St  Table Data
-----
mvr          Tagged 2220  1/a/8/0/eth  ethernet8-2220/bridge  DWN S MVR vlan 2220
upl          Tagged 2800  1/a/8/0/eth  ethernet8-2800/bridge  DWN S VLAN 2800 default
upl          Tagged 3800  1/a/8/0/eth  ethernet8-3800/bridge  DWN S VLAN 3800 default
3 Bridge Interfaces displayed
```

3 Create the downlink bridges on the subscriber facing Ethernet ports for both MVR and video.

The VLAN ID passes all traffic that is not downstream multicast traffic and the MVR VLAN passes the multicast video traffic.

Multicast streams for video will enter the downlink bridge on the MVR VLAN 2220.

```
zSH> bridge add 1-1-6-0/eth downlink-video vlan 2800 mvrvlan 2220 tagged video 0/3
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth-2800/bridge
```

```
zSH> bridge add 1-1-7-0/eth downlink-video vlan 3800 mvrvlan 2220 tagged video 0/2
Adding bridge on 1-1-7-0/eth
Created bridge-interface-record 1-1-7-0-eth-3800/bridge
```

Verify the bridges.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
dwn-vid    Tagged 2800  1/1/6/0/eth  1-1-6-0-eth-2800/bridge  UP
dwn-vid    Tagged 3800  1/1/7/0/eth  1-1-7-0-eth-3800/bridge  DWN
mvr        Tagged 2220  1/a/8/0/eth  ethernet8-2220/bridge    DWN S MVR vlan 2220
upl        Tagged 2800  1/a/8/0/eth  ethernet8-2800/bridge    DWN S VLAN 2800 default
upl        Tagged 3800  1/a/8/0/eth  ethernet8-3800/bridge    DWN S VLAN 3800 default
5 Bridge Interfaces displayed
```

Deleting single-tagged to single-tagged bridged video with MVR

1 Verify the bridges.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St  Table Data
-----
dwn-vid    Tagged 2800  1/1/6/0/eth  1-1-6-0-eth-2800/bridge  UP
dwn-vid    Tagged 3800  1/1/7/0/eth  1-1-7-0-eth-3800/bridge  DWN
mvr        Tagged 2220  1/a/8/0/eth  ethernet8-2220/bridge    DWN S MVR vlan 2220
upl        Tagged 2800  1/a/8/0/eth  ethernet8-2800/bridge    DWN S VLAN 2800 default
upl        Tagged 3800  1/a/8/0/eth  ethernet8-3800/bridge    DWN S VLAN 3800 default
5 Bridge Interfaces displayed
```

2 Delete the MVR bridge on the Ethernet uplink port.

```
zSH> bridge delete ethernet8-2220/bridge vlan 2220
Bridge-path deleted successfully
ethernet8-2220/bridge delete complete
```

3 Delete the uplink bridges on the Ethernet uplink port.

```
zSH> bridge delete ethernet8-2800/bridge vlan 2800
Bridge-path deleted successfully
ethernet8-2800/bridge delete complete
```

```
zSH> bridge delete ethernet8-3800/bridge vlan 3800
Bridge-path deleted successfully
ethernet8-3800/bridge delete complete
```

4 Delete the downlink bridges on the subscriber facing Ethernet ports.

```
zSH> bridge delete 1-1-6-0-eth-2800/bridge vlan 2800
1-1-6-0-eth-2800/bridge delete complete
```

```
zSH> bridge delete 1-1-7-0-eth-3800/bridge vlan 3800
1-1-7-0-eth-3800/bridge delete complete
```

Bridged video on the MXK with VLAN translation and MVR

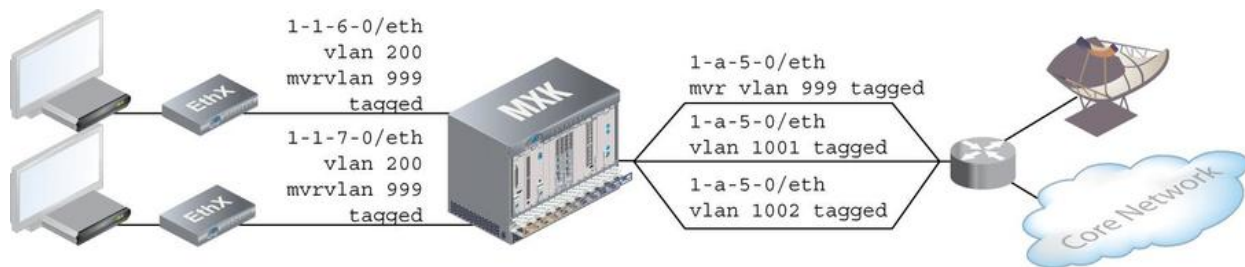
This section describes configuring asymmetric bridges on the MXK for video, VLAN translation, and MVR for IGMP.

When the downstream CPEs are pre-configured with the same VLAN ID, the downlink bridges can be configured so that the MXK translates the VLAN ID to a different VLAN ID for the uplink.

When configuring a bridge for MVR video, you create an MVR bridge for the downstream multicast video, and uplink bridges for everything that is not downstream multicast video and upstream IGMP. You create downlink bridges for VLAN translation, video, and to receive MVR.

MVR bridges are always *tagged*. Any bridge that passes multicast IP video traffic *must* be *tagged*.

Figure 59: Asymmetric bridge configuration with MVR and VLAN translation



Configuring single-tagged to single-tagged asymmetric bridges for VLAN translation and MVR

When configuring a bridge for video with MVR, you create an MVR bridge for the downstream multicast, and uplink bridges for everything that is not downstream multicast video and upstream IGMP.

In this single-tagged to single-tagged configuration, all bridges: MVR, uplink, and downlink are *tagged*.

Any bridge that passes multicast traffic *must* be *tagged*.

- 1 Create a *tagged* MVR bridge with VLAN ID on an Ethernet uplink port.

```
zSH> bridge add 1-a-5-0/eth mvr vlan 999 tagged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-999/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
mvr           Tagged 999  1/a/5/0/eth  ethernet5-999/bridge  UP  S MVR vlan 999
1 Bridge Interfaces displayed
```

View the bridge path.

```
zSH> bridge-path show
VLAN/SLAN   Bridge                               Address
-----
          999 ethernet5-999/bridge      MVR, MCAST Age: 250, IGMP Query Interval: 120,
IGMP Proxy, IGMP DSCP: 0
```

The defaults for video created on MVR bridge paths are:

- IGMP proxy reporting is enabled and sends the default IP address 10.10.10.0
- **mcast** is set to 250 seconds
- **igmptimer** is set to 120 seconds



Note: If your network checks for the source IP addresses, the default proxy IP address can be configured to a custom IP address.

igmpsendip is set to *enable* with the custom IP address.

Configure the bridge path with a custom IP address for proxy.

```
zSH> bridge-path modify ethernet5-999/bridge vlan 999 mvr igmpsendip enable
172.16.24.1
Bridge-path ethernet5-999/bridge/13/999/0/0/0/0/0/0/0 has been modified
```

Verify the bridge path.

```
zSH> bridge-path show
VLAN/SLAN   Bridge                               Address
-----
          999 ethernet5-999/bridge      MVR, MCAST Age: 241, IGMP Query Interval: 120,
IGMP Proxy, Custom IP 172.16.24.1, IGMP DSCP: 0
```

2 Create *tagged* uplink bridges with VLAN ID.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 1001 tagged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-1001/bridge
Bridge-path added successfully
```

```
zSH> bridge add 1-a-5-0/eth uplink vlan 1002 tagged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-1002/bridge
Bridge-path added successfully
```

Verify the bridges.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical      Bridge
-----
mvr      Tagged  999    1/a/5/0/eth  ethernet5-999/bridge  UP S MVR vlan 999
upl      Tagged  1001   1/a/5/0/eth  ethernet5-1001/bridge UP S VLAN 1001 default
```

```
upl          Tagged 1002 1/a/5/0/eth          ethernet5-1002/bridge          UP S VLAN 1002 default
3 Bridge Interfaces displayed
```

3 Create downlinks for to receive MVR with VLAN ID translation.

```
zSH> bridge add 1-1-6-0/eth downlink-video vlan 200 xlate-to 1001 mvrvlan 999
tagged video 0/3
```

```
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth-1001/bridge
```

```
zSH> bridge add 1-1-7-0/eth downlink-video vlan 200 xlate-to 1002 mvrvlan 999
tagged video 0/3
```

```
Adding bridge on 1-1-7-0/eth
Created bridge-interface-record 1-1-7-0-eth-1002/bridge
```

Verify the bridges.

```
zSH> bridge show
```

Type	Orig VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-vid	200	Tagged 1001	1/1/6/0/eth	1-1-6-0-eth-1001/bridge	UP	D 00:02:71:2e:2b:61
dwn-vid	200	Tagged 1002	1/1/7/0/eth	1-1-7-0-eth-1002/bridge	DWN	
mvr		Tagged 999	1/a/5/0/eth	ethernet5-999/bridge	DWN	S MVR vlan 999
upl		Tagged 1001	1/a/5/0/eth	ethernet5-1001/bridge	DWN	S VLAN 1001 default
upl		Tagged 1002	1/a/5/0/eth	ethernet5-1002/bridge	DWN	S VLAN 1002 default

5 Bridge Interfaces displayed

Deleting the single-tagged to single-tagged VLAN translation with MVR configuration

1 View the bridges.

```
zSH> bridge show
```

Type	Orig VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-vid	200	Tagged 1001	1/1/6/0/eth	1-1-6-0-eth-1001/bridge	UP	D 00:02:71:2e:2b:61
dwn-vid	200	Tagged 1002	1/1/7/0/eth	1-1-7-0-eth-1002/bridge	DWN	
mvr		Tagged 999	1/a/5/0/eth	ethernet5-999/bridge	DWN	S MVR vlan 999
upl		Tagged 1001	1/a/5/0/eth	ethernet5-1001/bridge	DWN	S VLAN 1001 default
upl		Tagged 1002	1/a/5/0/eth	ethernet5-1002/bridge	DWN	S VLAN 1002 default

5 Bridge Interfaces displayed

2 Delete the uplink bridge with MVR for multicast.

```
zSH> bridge delete ethernet5-999/bridge vlan 999
```

```
Bridge-path deleted successfully
ethernet5-999/bridge delete complete
```

3 Delete the uplink bridges for all other traffic.

```
zSH> bridge delete ethernet5-1001/bridge vlan 1001
```

```
Bridge-path deleted successfully
ethernet5-1001/bridge delete complete
```

```
zSH> bridge delete ethernet5-1002/bridge vlan 1002
Bridge-path deleted successfully
ethernet5-1002/bridge delete complete
```

- 4 Delete the downlink bridges. Bridges with VLAN ID translation use the translated VLAN ID in the **bridge delete** syntax.



Note: The VLAN ID added is *different* from the VLAN ID deleted.

```
zSH> bridge delete 1-1-6-0-eth-1001/bridge vlan 1001
1-1-6-0-eth-1001/bridge delete complete
```

```
zSH> bridge delete 1-1-7-0-eth-1002/bridge vlan 1002
1-1-7-0-eth-1002/bridge delete complete
```

Bridged video on the MXK with SLAN promotion and MVR

This section describes configuring asymmetric bridges on the MXK for video, SLAN promotion, and MVR for IGMP.

In this configuration, the MVR bridge is *tagged*, the uplink bridge is *staged*, and the downlink bridge is *tagged*.

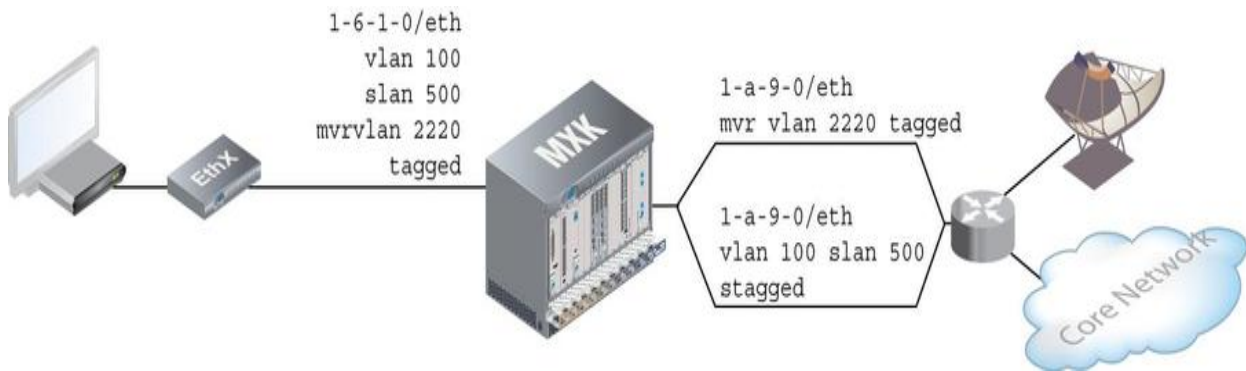
As shown in [Figure 60](#), the uplink bridge passes the VLAN ID to the network and the SLAN ID is promoted to the network, the downlink bridge passes the VLAN ID down to the subscriber's CPE and the subscriber receives multicast video traffic from the MVR bridge with MVR VLAN ID.

When a core network device is expecting a double-tagged configuration, (SLAN ID), a SLAN ID can be added from the downlink configuration to be promoted to the uplink. In this case, because the downlink bridge is *tagged*, the SLAN ID is not sent downstream. The uplink bridge is *staged* so the SLAN ID is sent to the network.

When configuring a bridge for MVR video, you create an MVR bridge for the downstream multicast video, and uplink bridges for everything that is not downstream multicast video and upstream IGMP. You create downlink bridges for MVR, video, and in this case, SLAN promotion.

MVR bridges are always *tagged*. Any bridge that passes multicast traffic *must* be *tagged*.

Figure 60: Asymmetric bridges with SLAN promotion and MVR



Creating asymmetric bridges for SLAN promotion and MVR

- 1 Create the MVR bridge on a network facing Ethernet port with the MVR VLAN ID for downstream multicast video traffic.

```
zSH> bridge add 1-a-9-0/eth mvr vlan 1111 tagged
Adding bridge on 1-a-9-0/eth
Created bridge-interface-record ethernet9-1111/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
mvr          Tagged  1111      1/a/9/0/eth  ethernet9-1111/bridge  DWN S MVR vlan 1111
1 Bridge Interfaces displayed
```

Verify the bridge path.

```
zSH> bridge-path show
VLAN/SLAN  Bridge  Address
-----
1111 ethernet9-1111/bridge  MVR, MCAST Age: 250, IGMP Query Interval: 120,
IGMP Proxy, IGMP DSCP: 0
```

The defaults for video created on MVR bridge paths are:

- IGMP proxy reporting is enabled and sends the default IP address 10.10.10.0
- **mcast** is set to 250 seconds
- **igmptimer** is set to 120 seconds



Note: If your network checks for the source IP addresses, the default proxy IP address can be configured to a custom IP address.

igmpsendip is set to *enable* with the custom IP address.

Configure the bridge path with a custom IP address for proxy.

```
zSH> bridge-path modify ethernet9-1111/bridge vlan 1111 mvr igmpsendip enable
172.16.24.1
Bridge-path ethernet9-1111/bridge/13/1111/0/0/0/0/0/0/0 has been modified
```

View the bridge path.

```
zSH> bridge-path show
VLAN/SLAN   Bridge                                     Address
-----
1111 ethernet9-1111/bridge               MVR, MCAST Age: 241, IGMP Query Interval: 120,
IGMP Proxy, Custom IP 172.16.24.1, IGMP DSCP: 0
```

- 2 Create the *tagged* uplink bridge for all traffic other than downstream multicast traffic with VLAN ID and SLAN ID.

```
zSH> bridge add 1-a-9-0/eth uplink vlan 100 slan 500 tagged
Adding bridge on 1-a-9-0/eth
Created bridge-interface-record ethernet9-100-500/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical      Bridge                                     St Table Data
-----
upl      ST 100/500   1/a/9/0/eth  ethernet9-100-500/bridge                DWN S SLAN 500 VLAN 100 default
mvr      Tagged 1111  1/a/9/0/eth  ethernet9-1111/bridge                    DWN S MVR vlan 1111
2 Bridge Interfaces displayed
```

- 3 Create the *tagged* downlink bridge to receive MVR, SLAN promotion, and video.

```
zSH> bridge add 1-1-6-0/eth downlink-video vlan 100 slan 500 mvrvlan 1111 tagged
video 0/3
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth-100/bridge
```

Verify the bridge.

```
zSH> bridge show
Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical      Bridge                                     St Table Data
-----
dwn-vid  Tg 100/500   1/1/6/0/eth  1-1-6-0-eth-100/bridge                UP
upl      ST 100/500   1/a/9/0/eth  ethernet9-100-500/bridge                DWN S SLAN 500 VLAN 100 default
mvr      Tagged 1111  1/a/9/0/eth  ethernet9-1111/bridge                    DWN S MVR vlan 1111
3 Bridge Interfaces displayed
```

Deleting bridges for SLAN promotion and MVR

- 1 Verify the bridges.

```
zSH> bridge show
Orig
```


Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St Table Data
dwn-vid		Tg 100/500	1/1/6/0/eth	1-1-6-0-eth-100/bridge	UP
upl		ST 100/500	1/a/9/0/eth	ethernet9-100-500/bridge	DWN S SLAN 500 VLAN 100 default
mvr		Tagged 1111	1/a/9/0/eth	ethernet9-1111/bridge	DWN S MVR vlan 1111
3 Bridge Interfaces displayed					

2 Delete the MVR bridge with VLAN ID.

```
zSH> bridge delete ethernet9-1111/bridge vlan 1111
Bridge-path deleted successfully
ethernet9-1111/bridge delete complete
```

3 Delete the uplink bridge.

```
zSH> bridge delete ethernet9-100-500/bridge vlan 100
Bridge-path deleted successfully
ethernet9-100-500/bridge delete complete
```

4 Delete the downlink bridge.

```
zSH> bridge delete 1-1-6-0-eth-100/bridge vlan 100
1-1-6-0-eth-100/bridge delete complete
```

Bridged video on the MXK with VLAN translation, SLAN promotion, and MVR

This section describes configuring asymmetric bridges on the MXK for video, VLAN translation, SLAN promotion, and MVR for IGMP.

When the downstream CPEs are pre-configured with the same VLAN ID, the downlink bridges can be configured to translate the common VLAN ID to different VLAN IDs on the uplink.

When a core network device is also expecting an SLAN ID, an SLAN ID can be added to the downlink configuration to be promoted to the uplink.

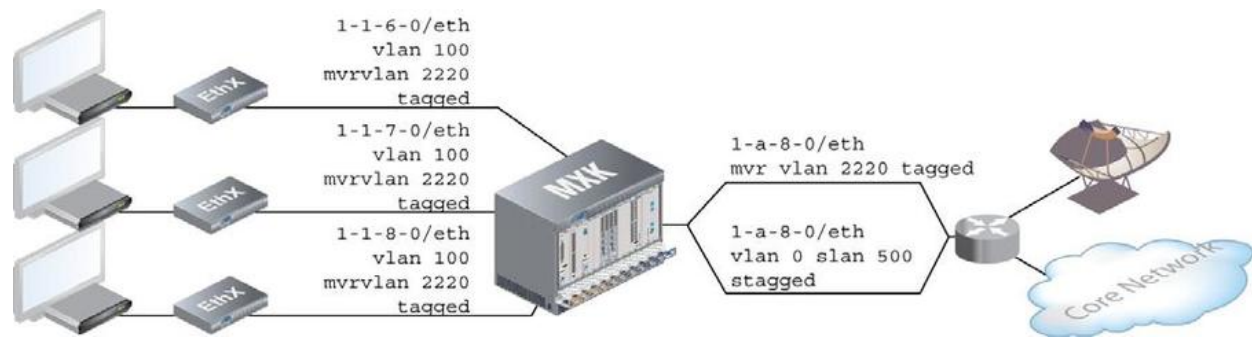
In this case, because the downlink bridge is *tagged*, the SLAN ID is not sent downstream and the uplink bridge is *staged* to send the SLAN ID to the network.

When configuring a bridge for MVR video, you create an MVR bridge for downstream multicast video, and uplink bridges for everything that is not downstream multicast video and upstream IGMP. You create downlink bridges for VLAN translation, video, and SLAN promotion.

MVR bridges are always *tagged*. Any bridge that passes multicast traffic *must* be *tagged*.

As shown in [Figure 61](#), the uplink bridge passes the VLAN ID to the network and the SLAN ID is promoted to the network, the downlink bridge passes the VLAN ID down to the subscriber's CPE and the subscriber receives multicast video traffic from the MVR bridge with the MVR VLAN ID.

Figure 61: Asymmetric bridge configuration with VLAN translation, SLAN promotion, and MVR



Creating asymmetric bridges for MVR, VLAN translation, and SLAN promotion

When configuring a bridge for video with MVR, you create an MVR bridge for the downstream multicast, and an uplink bridge for everything that is not downstream multicast.

- 1 Create a *tagged* MVR bridge with VLAN ID on an uplink Ethernet port.

```
zSH> bridge add 1-a-8-0/eth mvr vlan 2220 tagged
Adding bridge on 1-a-8-0/eth
Created bridge-interface-record ethernet8-2220/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN      Physical      Bridge      St  Table Data
-----
mvr      Tagged 2220  1/a/8/0/eth    ethernet8-2220/bridge  DWN S MVR vlan 2220
1 Bridge Interfaces displayed
```

Verify the automatically created bridge path.

```
zSH> bridge-path show
VLAN/SLAN  Bridge      Address
-----
2220 ethernet8-2220/bridge  MVR, MCAST Age: 250, IGMP Query
Interval: 120, IGMP Proxy, IGMP DSCP: 0
```

The defaults for video created on MVR bridge paths are:

- IGMP proxy reporting is enabled and sends the default IP address 10.10.10.0
- **mcast** is set to 250 seconds
- **igmp timer** is set to 120 seconds



Note: If your network checks for the source IP addresses, the default proxy IP address can be configured to a custom IP address.

igmpsendip is set to *enable* with the custom IP address.

Configure the bridge path with a custom IP address for proxy.

```
zSH> bridge-path modify ethernet8-2220/bridge vlan 2220 mvr igmpsendip enable
172.16.24.1
Bridge-path ethernet8-2220/bridge/13/2220/0/0/0/0/0/0/0 has been modified
```

Verify the bridge path.

```
zSH> bridge-path show
```

VLAN/SLAN	Bridge	Address
2220 ethernet8-2220/bridge	MVR, MCAST Age: 241, IGMP Query Interval: 120, IGMP Proxy, Custom IP 172.16.24.1, IGMP DSCP: 0	

2 Create the uplink bridge with VLAN ID 0 (accepts all VLANs) and SLAN ID 500 *tagged*.

This uplink accepts all VLAN IDs, passes the VLAN ID to the network and promotes the SLAN ID to the network.

```
zSH> bridge add 1-a-8-0/eth uplink vlan 0 slan 500 tagged
Adding bridge on 1-a-8-0/eth
Created bridge-interface-record ethernet8-0-500/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
```

Type	VLAN/SLAN	Orig	Physical	Bridge	St	Table Data
upl	ST 0/500	1/a/8/0/eth	ethernet8-0-500/bridge	DWN	S	SLAN 500 VLAN 0 default
mvr	Tagged 2220	1/a/8/0/eth	ethernet8-2220/bridge	DWN	S	MVR vlan 2220

2 Bridge Interfaces displayed

3 Create the downlink bridges to receive MVR, for VLAN translation and SLAN promotion, and video.

```
zSH> bridge add 1-1-6-0/eth downlink-video vlan 100 xlate-to 1001 slan 500 mvrvlan
2220 tagged video 1/2
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth-1001-500/bridge
```

```
zSH> bridge add 1-1-7-0/eth downlink-video vlan 100 xlate-to 1002 slan 500 mvrvlan
2220 tagged video 1/3
Adding bridge on 1-1-7-0/eth
Created bridge-interface-record 1-1-7-0-eth-1002-500/bridge
```

```
zSH> bridge add 1-1-8-0/eth downlink-video vlan 100 xlate-to 1003 slan 500 mvrvlan
2220 tagged video 1/3
```

Adding bridge on 1-1-8-0/eth
 Created bridge-interface-record 1-1-8-0-eth-1003-500/bridge

Verify the bridges.

```
zSH> bridge show
Orig
Type      VLAN/SLAN  VLAN/SLAN  Physical      Bridge                                     St  Table Data
-----
dwn-vid   100/----  Tg 1001/500  1/1/6/0/eth   1-1-6-0-eth-1001-500/bridge              UP
dwn-vid   100/----  Tg 1002/500  1/1/7/0/eth   1-1-7-0-eth-1002-500/bridge              DWN
dwn-vid   100/----  Tg 1003/500  1/1/8/0/eth   1-1-8-0-eth-1003-500/bridge              DWN
upl                ST   0/500  1/a/8/0/eth   ethernet8-0-500/bridge                   DWN S SLAN 500 VLAN 0 default
mvr                Tagged 2220  1/a/8/0/eth   ethernet8-2220/bridge                   DWN S MVR vlan 2220

5 Bridge Interfaces displayed
```

Deleting the single-tagged to double-tagged bridges with MVR

- 1 Delete the downlink bridges. Downlink bridges with VLAN translation use the translated VLAN ID in the **bridge delete** syntax.



Note: The VLAN ID added is *different* from the VLAN ID deleted.

```
zSH> bridge delete 1-1-6-0-eth-1001-500/bridge vlan 1001
1-1-6-0-eth-1001-500/bridge delete complete
```

```
zSH> bridge delete 1-1-7-0-eth-1002-500/bridge vlan 1002
1-1-7-0-eth-1002-500/bridge delete complete
```

```
zSH> bridge delete 1-1-8-0-eth-1003-500/bridge vlan 1003
1-1-8-0-eth-1003-500/bridge delete complete
```

- 2 Delete the MVR bridge.

```
zSH> bridge delete ethernet8-2220/bridge vlan 2220
Bridge-path deleted successfully
ethernet8-2220/bridge delete complete
```

- 3 Delete the uplink bridge.

```
zSH> bridge delete ethernet8-0-500/bridge vlan 0
Bridge-path deleted successfully
ethernet8-0-500/bridge delete complete
```

Bridged video on the MXK with dual MVR

This section describes configuring asymmetric bridges on the MXK with dual MVR for IGMP and video and includes:

- [Bridged video with no MVR, page 457](#)
- [Bridged video with single MVR, page 457](#)

- [Bridged video with dual MVR, page 457](#)

The dual MVR feature allows for two uplink bridge interfaces to be associated to downlink bridge interfaces.

When configuring the bridges for dual MVR video, you create two MVR bridges for the downstream multicast video, and uplink bridges for everything else that is not downstream multicast. You must also link the two MVR bridges with the **bridge-path add** command.

Bridged video with no MVR

In bridged video configurations with no MVR VLAN, a video VLAN *x* is configured on both the network facing uplink bridge and the subscriber facing downlink bridge. Video content arrives from the network on VLAN *x* and is multicast to all VLAN *x* downlinks. When the subscriber sends IGMP join requests to the network, that request is processed on VLAN *x*.

```
bridge add 1-a-7-0/eth uplink vlan x tagged igmpproxy
```

```
bridge add 1-4-1-701/gponport gtp 1 downlink vlan x
tagged video 0/3
```

Bridged video with single MVR

With single MVR, an MVR VLAN *y* is created on the network port and video content arrives from the network on MVR VLAN *y* allowing video subscribers to share one multicast VLAN in the network while remaining in their own unique subscriber VLAN, in this case VLAN *x*. The downlink configuration includes both VLAN *x* and MVR VLAN *y*. The MVR VLAN *y* is mapped to unique subscriber VLAN *x* before multicasting it downstream and IGMP join requests are mapped from VLAN *x* to VLAN *y* upstream.

```
bridge add 1-a-7-0/eth uplink vlan x tagged
```

```
bridge add 1-a-7-0/eth mvr vlan y tagged
```

```
bridge add 1-4-1-701/gponport gtp 1 downlink vlan x
mvrvlan y tagged video 0/3
```

Bridged video with dual MVR

With dual MVR, two MVR VLANs *y* and *z* are created for two separate video multicast streams, such as SD and HD, coming down from the network. MVR VLAN *y* and MVR VLAN *z* are mapped together on the uplink bridge interface with the **bridge-path add** command. Downstream, both MVR VLANs *y* and *z* are mapped to VLAN *x* on the downlink. This allows both video streams to map to the unique user VLAN *x* for multicast down and both IGMP join requests to be mapped to MVR VLANs *y* and *z* towards the network.

```
bridge add 1-a-7-0/eth uplink vlan x tagged
```

```
bridge add 1-a-7-0/eth mvr vlan y tagged

bridge add 1-a-7-0/eth mvr vlan z tagged

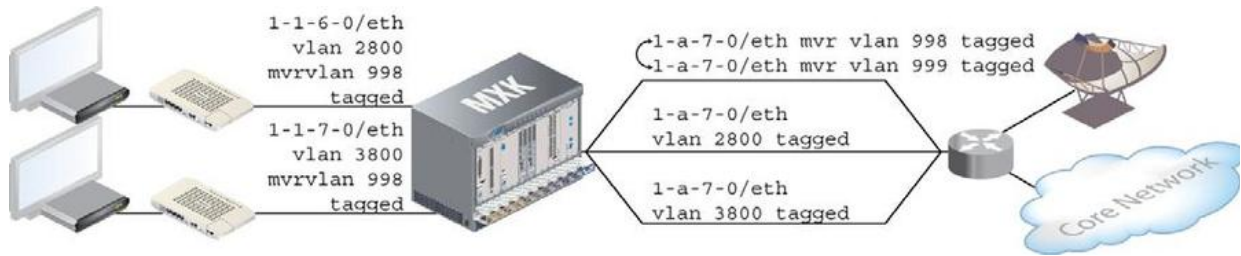
bridge-path add ethernet7-y/bridge vlan z secmvr

bridge add 1-6-1-0/eth downlink vlan x mvrvlan y tagged
video 0/3
```

MVR bridges are always *tagged*. Any bridge that passes multicast traffic *must* be *tagged*.

As shown in [Figure 58](#), the MVR bridge with MVR VLAN ID (after the two MVR bridges are mapped) can be used by multiple downlink bridges for downstream multicast video.

Figure 62: Asymmetric bridges with MVR and video



Creating single-tagged to single-tagged asymmetric bridged video with dual MVR

This case describes how dual MVR can be configured.

- 1 Create *tagged* uplink bridges for all traffic except the dual downstream multicast traffic.

```
zSH> bridge add 1-a-7-0/eth uplink vlan 2800 tagged
Adding bridge on 1-a-7-0/eth
Created bridge-interface-record ethernet7-2800/bridge
Bridge-path added successfully
```

```
zSH> bridge add 1-a-7-0/eth uplink vlan 3800 tagged
Adding bridge on 1-a-7-0/eth
Created bridge-interface-record ethernet7-3800/bridge
Bridge-path added successfully
```

Verify the bridges.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
upl          Tagged 2800  1/a/7/0/eth  ethernet7-2800/bridge  DWN S VLAN 2800 default
upl          Tagged 3800  1/a/7/0/eth  ethernet7-3800/bridge  DWN S VLAN 3800 default
2 Bridge Interfaces displayed
```

- 2 Create the first *tagged* MVR VLAN bridge on the same port as the uplink bridges for the first downstream multicast.

```
zSH> bridge add 1-a-7-0/eth mvr vlan 998 tagged
Adding bridge on 1-a-7-0/eth
Created bridge-interface-record ethernet7-998/bridge
Bridge-path added successfully
```

- 3 Create the second *tagged* MVR VLAN bridge on the same port as the uplink bridges for the second downstream multicast.

```
zSH> bridge add 1-a-7-0/eth mvr vlan 999 tagged
Adding bridge on 1-a-7-0/eth
Created bridge-interface-record ethernet7-999/bridge
Bridge-path added successfully
```

- 4 Verify the bridges and bridge paths. In this case both MVR VLAN IDs are displayed and two bridge paths are displayed.

```
zSH> bridge show
```

Orig							
Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
mvr	Tagged	998	1/a/7/0/eth	ethernet7-998/bridge	DWN	S	MVR vlan 998
mvr	Tagged	999	1/a/7/0/eth	ethernet7-999/bridge	DWN	S	MVR vlan 999
upl	Tagged	2800	1/a/7/0/eth	ethernet7-2800/bridge	DWN	S	VLAN 2800 default
upl	Tagged	3800	1/a/7/0/eth	ethernet7-3800/bridge	DWN	S	VLAN 3800 default

4 Bridge Interfaces displayed

```
zSH> bridge-path show
```

VLAN/SLAN	Bridge	Address
2800 ethernet7-2800/bridge	Default, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym	
3800 ethernet7-3800/bridge	Default, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym	
998 ethernet7-998/bridge	MVR, MCAST Age: 250, IGMP Query Interval: 120, IGMP Proxy, IGMP DSCP: 0	
999 ethernet7-999/bridge	MVR, MCAST Age: 250, IGMP Query Interval: 120, IGMP Proxy, IGMP DSCP: 0	

- 5 Map the two MVR VLAN IDs.

To map MVR VLANs enter the **bridge-path add** command, the bridge interface of one MVR VLAN bridge and the MVR VLAN of the other MVR VLAN bridge and the keyword **secmvr**. In this example, MVR VLAN 998 becomes the primary MVR VLAN, and MVR VLAN 999 is the secondary MVR VLAN.

```
zSH> bridge-path add ethernet7-998/bridge vlan 999 secmvr
Bridge-path added successfully
```

Verify the bridges and the bridge paths. The bridge interface and the bridge-path that is designated as the secondary MVR is now displayed, in this case MVR VLAN 999.

```
zSH> bridge show
Orig
```

Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
mvr	Tagged	998	1/a/7/0/eth	ethernet7-998/bridge	DWN S	MVR vlan 998 S Secondary MVR vlan 999
mvr	Tagged	999	1/a/7/0/eth	ethernet7-999/bridge	DWN S	MVR vlan 999
upl	Tagged	2800	1/a/7/0/eth	ethernet7-2800/bridge	DWN S	VLAN 2800 default
upl	Tagged	3800	1/a/7/0/eth	ethernet7-3800/bridge	DWN S	VLAN 3800 default

4 Bridge Interfaces displayed

zSH> **bridge-path show**

VLAN/SLAN	Bridge	Address
2800 ethernet7-2800/bridge	Default, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym	
3800 ethernet7-3800/bridge	Default, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0, IGMP DSCP: 0, Flap Mode: Default, Block: Asym	
998 ethernet7-998/bridge	MVR, MCAST Age: 250, IGMP Query Interval: 120, IGMP Proxy, IGMP DSCP: 0	
999 ethernet7-999/bridge	MVR, MCAST Age: 250, IGMP Query Interval: 120, IGMP Proxy, IGMP DSCP: 0	
999 ethernet7-998/bridge	Secondary MVR	

6 Create the downlink bridges on the subscriber facing Ethernet ports for both MVR and video. Enter the primary MVR VLAN.

zSH> **bridge add 1-1-6-0/eth downlink-video vlan 2800 mvrvlan 998 tagged video 0/3**
Adding bridge on 1-1-6-0/eth
Created bridge-interface-record 1-1-6-0-eth-2800/bridge

zSH> **bridge add 1-1-7-0/eth downlink-video vlan 3800 mvrvlan 998 tagged video 0/3**
Adding bridge on 1-1-7-0/eth
Created bridge-interface-record 1-1-7-0-eth-3800/bridge

Verify the bridges.

zSH> **bridge show**

Type	Orig	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn-vid		Tagged	2800	1/1/6/0/eth	1-1-6-0-eth-2800/bridge	UP	
dwn-vid		Tagged	3800	1/1/7/0/eth	1-1-7-0-eth-3800/bridge	DWN	
mvr		Tagged	998	1/a/7/0/eth	ethernet7-998/bridge	DWN S	MVR vlan 998 S Secondary MVR vlan 999
mvr		Tagged	999	1/a/7/0/eth	ethernet7-999/bridge	DWN S	MVR vlan 999
upl		Tagged	2800	1/a/7/0/eth	ethernet7-2800/bridge	DWN S	VLAN 2800 default
upl		Tagged	3800	1/a/7/0/eth	ethernet7-3800/bridge	DWN S	VLAN 3800 default

6 Bridge Interfaces displayed

Deleting single-tagged to single-tagged bridged video with MVR

1 Verify the bridges.

zSH> **bridge show**

Type	Orig	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data


```

-----
dwn-vid      Tagged 2800 1/1/6/0/eth      1-1-6-0-eth-2800/bridge      UP
dwn-vid      Tagged 3800 1/1/7/0/eth      1-1-7-0-eth-3800/bridge      DWN
mvr          Tagged 998  1/a/7/0/eth      ethernet7-998/bridge         DWN S MVR vlan 998
                                           S Secondary MVR vlan
999
mvr          Tagged 999  1/a/7/0/eth      ethernet7-999/bridge         DWN S MVR vlan 999
upl          Tagged 2800 1/a/7/0/eth      ethernet7-2800/bridge        DWN S VLAN 2800 default
upl          Tagged 3800 1/a/7/0/eth      ethernet7-3800/bridge        DWN S VLAN 3800 default
6 Bridge Interfaces displayed

```

- 2 Delete the downlink bridges on the subscriber facing ports. This will also delete the MVR mappings and allow the MVR bridges to be deleted.

```

zSH> bridge delete 1-1-6-0-eth-2800/bridge
1-1-6-0-eth-2800/bridge delete complete

```

```

zSH> bridge delete 1-1-7-0-eth-3800/bridge
1-1-7-0-eth-3800/bridge delete complete

```

- 3 Delete the uplink bridges on the network facing Ethernet ports.

```

zSH> bridge delete ethernet7-2800/bridge vlan 2800
Bridge-path deleted successfully
ethernet7-2800/bridge delete complete

```

```

zSH> bridge delete ethernet7-3800/bridge vlan 3800
Bridge-path deleted successfully
ethernet7-3800/bridge delete complete

```

- 4 Delete the dual MVR bridges on the Ethernet uplink port.

```

zSH> bridge delete ethernet7-998/bridge
Bridge-path deleted successfully
ethernet7-998/bridge delete complete

```

```

zSH> bridge delete ethernet7-999/bridge
ethernet7-999/bridge delete complete

```

Display bridge IGMP

This section describes:

- [Display bridge IGMP, page 462](#)
- [IGMP bridging statistics, page 463](#)
- [IGMPv3 and IGMPv2 proxy agent, page 465](#)

Display bridge IGMP

Displaying bridge IGMP

The **bridge igmp** command displays the time left for multicast when entered from the card slot, not the MXK system.



Note: The **bridge show** command on uplink bridges no longer displays multicast MAC addresses for the downlink bridges. Use the **bridge igmp slot <x>** command to display IGMP information.

To view IGMP data enter the **bridge igmp** command.

```
zSH> bridge igmp
```

Slot	Vlan	Join	Timer	MAC Address	MCAST IP	Bridge Host MAC	Last
0	998	01:00:5e:0a:63:0b		226.10.99.11		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:1a		226.10.99.26		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:21		226.10.99.33		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:28		226.10.99.40		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:2d		226.10.99.45		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:2e		226.10.99.46		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:33		226.10.99.51		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:37		226.10.99.55		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:3a		226.10.99.58		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:3c		226.10.99.60		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:3f		226.10.99.63		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:40		226.10.99.64		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:49		226.10.99.73		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:4b		226.10.99.75		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:4d		226.10.99.77		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:4e		226.10.99.78		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:50		226.10.99.80		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:54		226.10.99.84		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:61		226.10.99.97		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:62		226.10.99.98		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:65		226.10.99.101		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:66		226.10.99.102		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:71		226.10.99.113		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:77		226.10.99.119		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:79		226.10.99.121		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:82		226.10.99.130		Slot 7 00:01:47:00:00:07	3:44
0	998	01:00:5e:0a:63:8e		226.10.99.142		Slot 7 00:01:47:00:00:07	3:44

```

0 998 01:00:5e:0a:63:94 226.10.99.148 Slot 7 00:01:47:00:00:07 3:44
0 998 01:00:5e:0a:63:95 226.10.99.149 Slot 7 00:01:47:00:00:07 3:44

```

Or enter the **bridge igmp slot <#>** command:

```
zSH> bridge igmp slot 7
```

Slan	Vlan	MAC Address	MCAST IP	Bridge Host MAC	Last Join Timer
0	998	01:00:5e:0a:63:20	226.10.99.32	1-7-3-901-gponport-998	00:00:d0:00:00:01 4:10
0	998	01:00:5e:0a:63:2f	226.10.99.47	1-7-3-902-gponport-998	00:00:d0:00:00:02 3:22
0	998	01:00:5e:0a:63:66	226.10.99.102	1-7-3-903-gponport-998	00:00:d0:00:00:03 3:22
0	998	01:00:5e:0a:63:7f	226.10.99.127	1-7-3-904-gponport-998	00:00:d0:00:00:04 3:23
0	998	01:00:5e:0a:63:7b	226.10.99.123	1-7-3-905-gponport-998	00:00:d0:00:00:05 3:28
0	998	01:00:5e:0a:63:79	226.10.99.121	1-7-3-906-gponport-998	00:00:d0:00:00:06 3:26
0	998	01:00:5e:0a:63:13	226.10.99.19	1-7-3-907-gponport-998	00:00:d0:00:00:07 3:26
0	998	01:00:5e:0a:63:2a	226.10.99.42	1-7-3-908-gponport-998	00:00:d0:00:00:08 3:25
0	998	01:00:5e:0a:63:89	226.10.99.137	1-7-3-909-gponport-998	00:00:d0:00:00:09 3:28
0	998	01:00:5e:0a:63:5f	226.10.99.95	1-7-3-910-gponport-998	00:00:d0:00:00:0a 3:28
0	998	01:00:5e:0a:63:42	226.10.99.66	1-7-3-911-gponport-998	00:00:d0:00:00:0b 3:28
0	998	01:00:5e:0a:63:12	226.10.99.18	1-7-3-912-gponport-998	00:00:d0:00:00:0c 3:29
0	998	01:00:5e:0a:63:11	226.10.99.17	1-7-3-913-gponport-998	00:00:d0:00:00:0d 3:34
0	998	01:00:5e:0a:63:90	226.10.99.144	1-7-3-914-gponport-998	00:00:d0:00:00:0e 3:37
0	998	01:00:5e:0a:63:28	226.10.99.40	1-7-3-915-gponport-998	00:00:d0:00:00:0f 3:38
0	998	01:00:5e:0a:63:98	226.10.99.152	1-7-3-916-gponport-998	00:00:d0:00:00:10 3:42
0	998	01:00:5e:0a:63:11	226.10.99.17	1-7-3-917-gponport-998	00:00:d0:00:00:11 3:41
0	998	01:00:5e:0a:63:3d	226.10.99.61	1-7-3-918-gponport-998	00:00:d0:00:00:12 3:45
0	998	01:00:5e:0a:63:23	226.10.99.35	1-7-3-919-gponport-998	00:00:d0:00:00:13 3:48
0	998	01:00:5e:0a:63:8a	226.10.99.138	1-7-3-920-gponport-998	00:00:d0:00:00:14 3:49
0	998	01:00:5e:0a:63:5a	226.10.99.90	1-7-3-921-gponport-998	00:00:d0:00:00:15 3:56
0	998	01:00:5e:0a:63:0c	226.10.99.12	1-7-3-922-gponport-998	00:00:d0:00:00:16 3:56
0	998	01:00:5e:0a:63:91	226.10.99.145	1-7-3-923-gponport-998	00:00:d0:00:00:17 4:02
0	998	01:00:5e:0a:63:27	226.10.99.39	1-7-3-924-gponport-998	00:00:d0:00:00:18 4:01
0	998	01:00:5e:0a:63:69	226.10.99.105	1-7-3-925-gponport-998	00:00:d0:00:00:19 4:04
0	998	01:00:5e:0a:63:5c	226.10.99.92	1-7-3-926-gponport-998	00:00:d0:00:00:1a 4:09
0	998	01:00:5e:0a:63:78	226.10.99.120	1-7-3-927-gponport-998	00:00:d0:00:00:1b 4:09
0	998	01:00:5e:0a:63:54	226.10.99.84	1-7-3-928-gponport-998	00:00:d0:00:00:1c 3:16
0	998	01:00:5e:0a:63:54	226.10.99.84	1-7-3-929-gponport-998	00:00:d0:00:00:1d 3:18
0	998	01:00:5e:0a:63:18	226.10.99.24	1-7-3-930-gponport-998	00:00:d0:00:00:1e 3:19
0	998	01:00:5e:0a:63:17	226.10.99.23	1-7-3-931-gponport-998	00:00:d0:00:00:1f 3:23
0	998	01:00:5e:0a:63:4a	226.10.99.74	1-7-3-932-gponport-998	00:00:d0:00:00:20 3:24

In addition, you can run a **bridge igmp** command to determine whether IGMP is running on the system.

IGMP bridging statistics

Viewing IGMP bridge statistics



Note: The **ip igmpstat** command displays the ports receiving multicast traffic and the joined multicast group(s).

- 1 Entering the **bridge igmpstat vlan <x>** command displays IGMP information on the downlinks.

```
zSH> bridge igmpstat vlan 998
```

Interface		Received				Transmitted	
vxReport	Leave	GenQuery	SpecQuery	vxReport	Leave	GenQuery	SpecQuery
v2/v3	v2	v2/v3	v2/v3	v2/v3	v2	v2/v3	v2/v3
linkagg-14-1-998/bridge	0/0	0/0	0/0	0/0	0	687/2	0/0
linkagg-4-1-998/bridge	0/0	0/0	0/0	0/0	0	0/0	0/0
1-14-19-0-eth-998/bridge	0/0	0/0	0/0	0/0	0	687/2	0/0
1-14-20-0-eth-998/bridge	0/0	0/0	0/0	0/0	0	687/2	0/0
1-4-19-0-eth-998/bridge	0/0	0/0	0/0	0/0	0	687/2	0/0
1-14-1-0-eth-998/bridge	0/0	0/0	0/0	0/0	0	687/2	0/0
ethernet3-998/bridge	1699/0	1011/0	176/0	0/0	0	0/0	0/0
1-7-3-901-gponport-998/bridge	0/0	0/0	0/0	74/0	6	685/2	6/0
1-7-3-902-gponport-998/bridge	0/0	0/0	0/0	74/0	6	685/2	6/0
1-7-3-903-gponport-998/bridge	0/0	0/0	0/0	74/0	6	685/2	6/0
1-7-3-904-gponport-998/bridge	0/0	0/0	0/0	75/0	6	685/2	6/0
1-7-3-905-gponport-998/bridge	0/0	0/0	0/0	73/0	6	685/2	6/0
1-7-3-906-gponport-998/bridge	0/0	0/0	0/0	75/0	6	685/2	6/0
1-7-3-907-gponport-998/bridge	0/0	0/0	0/0	74/0	6	685/2	6/0
1-7-3-908-gponport-998/bridge	0/0	0/0	0/0	74/0	6	685/2	6/0
1-7-3-909-gponport-998/bridge	0/0	0/0	0/0	73/0	6	685/2	6/0
1-7-3-910-gponport-998/bridge	0/0	0/0	0/0	73/0	6	686/2	6/0
1-7-3-911-gponport-998/bridge	0/0	0/0	0/0	73/0	6	685/2	6/0
1-7-3-912-gponport-998/bridge	0/0	0/0	0/0	73/0	6	686/2	6/0
1-7-3-913-gponport-998/bridge	0/0	0/0	0/0	73/0	6	686/2	6/0
1-7-3-914-gponport-998/bridge	0/0	0/0	0/0	74/0	6	686/2	6/0
1-7-3-915-gponport-998/bridge	0/0	0/0	0/0	73/0	6	685/2	6/0
1-7-3-916-gponport-998/bridge	0/0	0/0	0/0	73/0	6	685/2	6/0
1-7-3-917-gponport-998/bridge	0/0	0/0	0/0	74/0	6	686/2	6/0
1-7-3-918-gponport-998/bridge	0/0	0/0	0/0	73/0	6	686/2	6/0
1-7-3-919-gponport-998/bridge	0/0	0/0	0/0	73/0	6	686/2	6/0
1-7-3-920-gponport-998/bridge	0/0	0/0	0/0	71/0	6	685/2	6/0

1-7-3-921-gponport-998/bridge 0/0 0	0/0	0/0	68/0	6	686/2	6/0
1-7-3-922-gponport-998/bridge 0/0 0	0/0	0/0	68/0	6	686/2	6/0
1-7-3-923-gponport-998/bridge 0/0 0	0/0	0/0	75/0	6	686/2	6/0
1-7-3-924-gponport-998/bridge 0/0 0	0/0	0/0	74/0	6	686/2	6/0
1-7-3-925-gponport-998/bridge 0/0 0	0/0	0/0	74/0	6	685/2	6/0
1-7-3-926-gponport-998/bridge 0/0 0	0/0	0/0	74/0	6	685/2	6/0
1-7-3-927-gponport-998/bridge 0/0 0	0/0	0/0	74/0	6	686/2	6/0
1-7-3-928-gponport-998/bridge 0/0 0	0/0	0/0	74/0	6	685/2	6/0
1-7-3-929-gponport-998/bridge 0/0 0	0/0	0/0	74/0	6	686/2	6/0
1-7-3-930-gponport-998/bridge 0/0 0	0/0	0/0	74/0	6	686/2	6/0
1-7-3-931-gponport-998/bridge 0/0 0	0/0	0/0	74/0	6	686/2	6/0
1-7-3-932-gponport-998/bridge 0/0 0	0/0	0/0	74/0	6	685/2	6/0

2 To view IGMP statistics information on the uplinks, enter the **bridge igmpstats slot <#> vlan <#>** command.

```
zSH> bridge igmpstats slot a vlan 998
Received                               Transmitted
Interface                               GenQuery SpecQuery vxReport Leave   GenQuery SpecQuery
vxReport Leave
v2/v3 v2/v3 v2/v3 v2 v2/v3 v2/v3 v2/v3 v2
ethernet3-998/bridge 1017/0 204/0 0/0 0 0/0 0/0
1909/0 204
1 Bridge Interfaces displayed
```

IGMPv3 and IGMPv2 proxy agent

This section describes

- [IGMPv3, page 465](#)
- [IGMPv2, page 466](#)

IGMPv3

This MXK release now supports IGMPv3 to the network and responds to the IGMPv3 messages. If an IGMP v2 query is received from the network the MXK's IGMP proxy agent will revert to v2 and continue using v2 until the next reboot or the IGMP version is reset with the **bridge igmpver reset** command.

IGMPv2

Configuring an uplink bridge for forced IGMPv2

If necessary, modify the uplink bridge-path to force IGMPv2 on the network and the subscriber side.

- 1 Create an uplink bridge for video with *igmpproxy*.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 100 igmpproxy
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-100/bridge
Bridge-path added successfully
```

- 2 Modify the uplink bridge-path to enable forced IGMPv2 on the network and the subscriber sides.

```
zSH> bridge-path modify ethernet2-100/bridge vlan 100 default forceigmpv2up enable
forceigmpv2down enable
Bridge-path ethernet2-100/bridge/3/100/0/0/0/0/0/0/0 has been modified
```

- 3 Verify the bridge-path.

```
zSH> bridge-path show
VLAN/SLAN  Bridge                                     Address
-----
100 ethernet2-100/bridge                     Default, Age: 3600, MCAST Age: 250,
IGMP Query Interval: 120, IGMP Proxy, IGMP DSCP: 0, forceIGMPv2Up,
forceIGMPv2Down, Flap Mode: Default, Block: Asym
```

- 4 If necessary, modify the uplink bridge-path to disable forced IGMPv2 on the network and the subscriber sides and revert to the default IGMPv3.

```
zSH> bridge-path modify ethernet2-100/bridge vlan 100 default forceigmpv2up
disable forceigmpv2down disable
Bridge-path ethernet2-100/bridge/3/100/0/0/0/0/0/0/0 has been modified
```

- 5 Verify the bridge-path.

```
zSH> bridge-path show
VLAN/SLAN  Bridge                                     Address
-----
100 ethernet2-100/bridge                     Default, Age: 3600, MCAST Age: 250,
IGMP Query Interval: 120, IGMP Proxy, IGMP DSCP: 0, Flap Mode: Default, Block: Asym
```

6

VOICE CONFIGURATION

This chapter describes the MXK voice cards and VoIP service configuration:

- [Voice cards, page 467](#)
- [VoIP configuration basic steps, page 468](#)
- [System settings, page 469](#)
- [Configure an IP interface for voice traffic, page 480](#)
- [Voice add command, page 481](#)
- [SIP, page 483](#)
- [SIP PLAR, page 500](#)
- [MGCP, page 507](#)
- [H.248, page 511](#)
- [Subscriber voice features configuration, page 522](#)
- [Advanced features, page 542](#)

Voice cards

The following MXK voice cards provide POTS VoIP services:

- MXK-POTS-72
- MXK-ADSL2+-POTS-BCM-48A-2S
- MXK-ADSL2+-POTS-BCM-48A-RNG-2S

Refer to [MXK POTS Cards, page 1345](#) for the detail.

The following MXK ISDN cards provide ISDN over packet voice service.

- MXK-ISDN-2B1Q-24
- MXK-ISDN-4B3T-24

VoIP configuration basic steps

These are the basic four steps to create the POTS to VoIP connection on MXK:

1. Set or verify that the system settings are appropriate.

Refer to [System settings on page 469](#).(Its one time setup)

2. Use the **interface add** command to create an IP interface.

Refer to [Configure an IP interface for voice traffic on page 480](#)



Note: IPv4 is supported for IP termination on the MXK. IPv6 is not supported for IP termination on the MXK.

3. Use the **new voip-server-entry** command to create the VoIP server.

This step configure the VoIP signaling protocols supported by the MXK:

The protocol setting can be configured as either Session Initiation Protocol (SIP) signaling, Media Gateway Control Protocol (MGCP), or H.248.

There is no need to create a voip server entry for SIP PLAR server (it gets automatically created when enter the **voice add plar** command.).



Note: MXK only supports one VoIP signaling protocol at a time, unless running ESA.



Caution: The system will automatically reboot if the voice signaling protocol is changed.

Refer to:

- [SIP on page 483](#)
- [SIP PLAR on page 500](#)
- [MGCP on page 507](#)
- [H.248 on page 511](#)

4. Use the **voice add** command to add the POTS to VoIP connection.

Refer to:

[Voice add command on page 481](#)

System settings

Before configuring a voice connection, make sure the system settings are configured to support the type of voice connection that you need.

The **system 0** profile contains settings that configure country-specific settings for voice calls and determines whether the system will reject incoming calls if there isn't enough bandwidth available.

Modifying the **countryregion** parameter of the **system** profile ensures that the country-specific voice settings are correctly set, such as voice encoding (A-law/Mu-law), ring-frequency, ring cadence, call progress tones, etc.

Certain voice settings on the voice card are designed for use in telephone systems located outside of North America. Refer to [Additional system settings on page 472](#) for where to modify some voice settings. For more information about those voice settings, contact your Zhone Technologies sales representative.

Setting a-law or mu-law and DSP settings

Modifying the **countryregion** parameter of the **system** profile ensures that the PCM encoding type (A-law/Mu-law) are correctly set. Mu-law is used in North America and Japan, and A-law used in most other countries.

The **show system** command displays the available system profile settings.

The A-law and Mu-law settings can also be set using the optional **alaw** and **mulaw** parameters in the **voice add** command.

For VoIP calls, if codec argument is not specified in the **voice add** command, the country code settings determines the default preferred-codec as g711mu or g711a.

Specifying a country with the same encoding type

This example changed **countryregion** from US to Canada in the **system** profile.

```
zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {Zhone Global Services and
Support 7195 Oakport Street Oakland Ca. (877) Zhone20
(946-6320) Fax (510)777-7113 support@zhone.com}:
sysname: -----> {Zhone MxK}:
syslocation: -----> {Oakland}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {false}:
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: ---> {false}:
```

```

configsyncpriority: ---> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: ---> {}:
configsyncstatus: -----> {syncinitializing}:
configsyncuser: -----> {}:
configsyncpasswd: -----> {** private **}: **
read-only **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {0.0.0.0}:
alternateipaddress: ---> {0.0.0.0}:
countryregion: -----> {us}: canada
primaryclocksource: ---> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: ->
{critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {NONE(0)}:
.....
Save changes? [s]ave, [c]hange or [q]uit:s

```

Record updated.

Specifying a country with the different encoding type

When you specify a country with a different encoding type, such as South Africa, in the **system** profile, you have the option of modifying the following dialing parameters in the **voice-system** profile:

- **hookflash-min-timer**
- **hookflash-max-timer**
- **pulse-inter-digit-timer**
- **min-make-pulse-width**
- **min-break-pulse-width**
- **max-break-pulse-width**

These options are read only after they have been set.



Note: After changing the countryregion to a country uses a different PCM encoding type, **reboot** system for this change to take effect.

- 1 To specify another country, such as South Africa, in the **system** profile:

```

zSH> update system 0
system 0

```

Please provide the following: [q]uit.

```

syscontact: -----> {Zhone Global Services and Support 7195 Oakport Street
Oakland Ca. (877) Zhone20 (946-6320) Fax (510)777-7113 support@zhone.com}:
sysname: -----> {Zhone MxK 5}:
syslocation: -----> {Oakland}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {true}:
zmsconnectionstatus: --> {active}:
zmsipaddress: -----> {172.16.48.89}:
configsyncexists: -----> {false}:
configsyncoverflow: ---> {false}:
configsyncpriority: ---> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: ---> {172.16.88.117_4_1280424907360}:
configsyncstatus: -----> {synccomplete}:
configsyncuser: -----> {zmsftp}:
configsyncpasswd: -----> {** private **}: ** read-only **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {172.16.88.117}:
alternateipaddress: ---> {0.0.0.0}:
countryregion: -----> {us}: southafrica
primaryclocksource: ---> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}:
webinterface: -----> {disabled}:
options: -----> {NONE(0)}:

```

.....

Save changes? [s]ave, [c]hange or [q]uit:**s**

countryregion changed to southafrica

Load country's pulse dialing parameters in voice-system profile ? [y]es or [n]o: **y**

voice-system profile updated with pulse dialing parameters for southafrica

sysMinBreakPulseWidth... 35 ms, sysMaxBreakPulseWidth... 75 ms

sysMinMakePulseWidth... 100 ms, sysPulseInterDigitTimer. 25 ms

minHookFlash..... 80 ms, maxHookFlash..... 230 ms

southafrica uses a different PCM encoding type (ALAW) from us (MULAW).

Please reboot the system for this change to take effect.

Record updated.

2 To verify or customize the country's pulse dialing parameters in voice-system profile:

```
zSH> update voice-system 0
```

```
voice-system 0
```

Please provide the following: [q]uit.

```
hookflash-min-timer: -----> {80}:
```

```
hookflash-max-timer: -----> {230}:
```

```

partial-dial-timeout: -----> {16}:
critical-dial-timeout: -----> {4}:
busy-tone-timeout: -----> {30}:
dial-tone-timeout: -----> {16}:
msg-wait-tone-timeout: -----> {16}:
offhook-warn-tone-timeout: -> {0}:
ringing-timeout: -----> {180}:
ringback-timeout: -----> {180}:
reorder-tone-timeout: -----> {30}:
stutter-tone-timeout: -----> {16}:
server-max-timer: -----> {20}:
config-max1: -----> {5}:
config-max2: -----> {7}:
max1-enable: -----> {true}:
max2-enable: -----> {true}:
max-waiting-delay: -----> {600}:
disconnection-wait-timer: --> {15}:
disconnection-min-timer: ---> {15}:
disconnection-max-timer: ---> {600}:
max-retransmit-timer: -----> {4}:
init-retransmit-timer: -----> {200}:
keep-alive-timer: -----> {60}:
no-response-timer: -----> {30}:
call-wait-max-repeat: -----> {2}:
call-wait-delay: -----> {10}:
pulse-inter-digit-timer: ---> {100}:
min-make-pulse-width: -----> {25}:
max-make-pulse-width: -----> {55}:
min-break-pulse-width: -----> {35}:
max-break-pulse-width: -----> {75}:
.....
Save changes? [s]ave, [c]hange or [q]uit:

```

Additional system settings

The following sections describe additional voice settings you might need to configure, depending on your network.

- [Specifying ring source, page 472](#)
- [Setting ring cadence and call progress parameters, page 473](#)
- [Changing the jitter buffer, page 477](#)
- [Configuring signal type and ring frequency, page 479](#)

Specifying ring source

By default, the system ring source is **internalringsourceable**, which means the system either use an on board MTAC/TAC card, or use a card with integrated ring generator (e.g. POTS 72 card) to get ring. You can also change the ring source to **externalringsourceable**, if the system use the external ring generator with an on board MTAC/TAC card.

This example changed ringsource from internalringsourcelabel to externalringsourcelabel in the **system** profile.

```

zSH> update system 0
system 0
Please provide the following: [q]uit.
syscontact: -----> {Zhone Global Services and Support 7195 Oakport
Street Oakland Ca. (877) Zhone20 (946-6320) Fax (510) 777-7113
support@zhone.com}:
sysname: -----> {Zhone MxK}:
syslocation: -----> {Oakland}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {false}:
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: ---> {false}:
configsyncpriority: ---> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: ---> {}:
configsyncstatus: -----> {syncinitializing}:
configsyncuser: -----> {}:
configsyncpasswd: -----> {** private **}: ** read-only **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {0.0.0.0}:
alternateipaddress: ---> {0.0.0.0}:
countryregion: -----> {us}:
primaryclocksource: ---> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}: externalringsourcelabel
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {NONE(0)}:
.....
Save changes? [s]ave, [c]hange or [q]uit:s
Record updated.

```

Setting ring cadence and call progress parameters

The MXK enables the ring cadence and other call progress parameters to be set for customized signal timing for SIP, MGCP, and H.248 calls.

For SIP systems, normal ring cadence or ring splash are used. For SIP PLAR systems, the class 5 switch determines the ring cadences, directly for GR303 and indirectly for V5.2 calls. For MGCP and H.248 systems, The MGCP and H.248 switches determine which ring cadence to use.

By default, ring cadences are set to standard United States settings. For Japan, other ring cadences are used that are not user-configurable. For other country-specific ring cadences, manually configure the ring cadences R0-R7 based on the country's requirements.

Table 38 lists the parameters that can be set. The following types of alert signal are used for on-hook signaling to wake up the caller ID device:

- **During Ringing**
The first ring is the alert signal, meaning the caller ID device is woken up to receive CLID data, when MXK provides the first ring.
- **Prior Ring with Dual Tone (DT) Wake Up (WU)**
A particular dual tone (2130Hz+2750Hz for 100ms) wakes up the caller ID CPE device for caller ID transmission. The tone and the caller ID signal are sent to prior to ringing.
- **Prior Ring with Ring Pulse (RP) Wake Up (WU)**
A short ring pulse (between 200ms and 300ms) wakes up the caller ID CPE device. Then, the caller ID signal transmission follows.
- **Prior Ring with Line Reversal (LR) Wake Up (WU)**
A line reversal (polarity change in DC voltage of the line, wakes up the caller ID device. Then, the caller ID signal transmission follows.
- **No Ring with Dual Tone (DT) Wake Up (WU)**
A particular dual tone (2130Hz+2750Hz for 100ms) wakes up the caller ID CPE device for caller ID transmission. Not associated with ringing.
- **No Ring with Ring Pules (RP) Wake Up (WU)**
A short ring pulse (between 200ms and 300ms) wakes up the caller ID CPE device. Not associated with ringing.
- **No Ring with Line Reversal (LR) Wake Up (WU)**
A line reversal (polarity change in DC voltage of the line, wakes up the caller ID device. Not associated with ringing.

Table 38: Ring cadence and call progress parameters

Parameter	Description
callerid-dig-protocol	Identifies the subscriber line protocol used for signaling on-hook caller id information. Different countries define different caller id signaling protocols to support caller identification. Supported protocols are Frequency Shift Keying (FSK) and Dual-Tone Multi-Frequency (DTMF).
r0-ring-cadence to r7-ring-cadence	Customized ring cadences. Ring cadence is required for the L line package.
ring cadence	Normal ring cadence

Table 38: Ring cadence and call progress parameters (Continued)

Parameter	Description
ring-splash-cadence	
power-ring frequency	the frequency at which the sinusoidal voltage must travel down the twisted pair to make terminal equipment ring. Different countries define different electrical characteristics to make terminal equipment ring. The f##Hz setting corresponds to a power ring frequency of ## Hertz. For example, the f25Hz setting corresponds to a power ring frequency of 25 Hertz. The f33Point33Hz setting corresponds to a power ring frequency of 33.33 Hertz.
clid-mode	The method of caller ID for on-hook caller ID. The Frequency Shift Keying (FSK) containing the Caller ID information is sent between the first and second ring pattern. For the dtas, rpas, and lr methods, the FSK containing the Caller ID information is sent before the first ring pattern. For the dtas method, the FSK is sent after the Dual Tone Alert Signal. For the rpas method, the FSK is sent after a Ring Pulse. For the lr method, the Line Reversal occurs first, then the Dual Tone Alert Signal, and finally the FSK is sent.
delay-before-clid-after-ring	The delay between the first ringing pattern and the start of the transmission of the FSK containing the Caller ID information. It is only used when CIDMode is duringRingingETS. The default value is 550 ms.
delay-before-clid-after-dtas	The delay between the end of the Dual Tone Alert Signal (DT-AS) and the start of the transmission of the FSK containing the Caller ID information. It is only used when CIDMode is dtas or lr. The default value is 50 ms.
delay-before-clid-after-rpas	The delay between the end of the Ring Pulse Alert Signal (RP-AS) and the start of the transmission of the FSK containing the Caller ID information. It is only used when CIDMode is rpas. The default value is 650 ms.
delay-after-clid-before-ring	The delay between the end of the complete transmission of the FSK containing the Caller ID information and the start of the first ring pattern. It is only used when CIDMode is dtas, rpas or lr. The default value is 250 ms.
delay-before-dtas-after-lr	The delay between the end of the Line Reversal and the start of the Dual Tone Alert Signal (DT-AS). It is only used when CIDMode is lr. The default value is 250 ms.
delay-before-vmwi-after-dtas	The delay between the end of the Dual Tone Alert Signal (DT-AS) and the start of the transmission of the FSK containing the VMWI information. It is only used when VmwiMode is dtas or lr. The default is 50 ms.
delay-before-vmwi-after-rpas	The delay between the end of the Ring Pulse Alert Signal (RP-AS) and the start of the transmission of the FSK containing the VMWI information. It is only used when VmwiMode is rpas. The default is 650 ms.

Table 38: Ring cadence and call progress parameters (Continued)

Parameter	Description
vmwi-delay-before-dtas-after-lr	The delay between the end of the Line Reversal and the start of the Dual Tone Alert Signal (DT-AS) for VMWI information. It is only used when VmwiMode is lr. The default is 250 ms.

In certain specific situations it may be necessary to reduce the length of the ring timer. The length of the ring timer can be adjusted in the **voice-call-process-config** profile.

The MXK automatically cuts off ringing if the ringing exceeds 2.2s. To configure the ringing cutoff timer, it can be done by changing any of the ring cadence fields in the **voice-call-progress-config** profile.

The format for ring cadence fields is **rec-x:on-y:off**.

where

- **rec** indicates the recursive nature of the cadence (continuous repeat of the same pattern).
 - “r” for recursive
 - “nr” for non-recursive
- **x:on** indicates to ring ON for x milliseconds.
- **y:off** indicates to ring OFF for x milliseconds.

For example, **r-2000:on-4000:off** indicates that the cadence is recursive with 2000msec ring on and 4000msec ring off cadence.

The **voice-call-process-config** profile configures all the voice call processing in a system.

The following examples changes ring cadence r0 and r1 from two seconds on, four seconds off in a repeating pattern to two seconds on, three seconds off, also in a repeating pattern.

Update the voice-call-process-config profile.

```
zSH> update voice-call-progress-config 0
```

```
voice-call-progress-config 0
Please provide the following: [q]uit.
callerid-sig-protocol: -----> {fsk}:
r0-ring-cadence: -----> {r-2000:on-4000:off}:
r-2000:on-3000:off
r1-ring-cadence: -----> {r-2000:on-4000:off}:
r-2000:on-3000:off
r2-ring-cadence: -----> {r-800:on-400:off-800:on-4000:off}:
r3-ring-cadence: ----->
{r-400:on-200:off-400:on-200:off-800:on-4000:off}:
r4-ring-cadence: ----->
{r-300:on-200:off-1000:on-200:off-300:on-4000:off}:
r5-ring-cadence: -----> {nr-500:on}:
```



```

r6-ring-cadence: -----> {r-2000:on-4000:off}:
r7-ring-cadence: -----> {r-2000:on-4000:off}:
ring-cadence: -----> {r-2000:on-4000:off}:
ring-splash-cadence: -----> {nr-500:on}:
power-ring-frequency: -----> {f20hz}:
clid-mode: -----> {duringringingets}:
delay-before-clid-after-ring: ----> {550}:
delay-before-clid-after-dtas: ----> {50}:
delay-before-clid-after-rpas: ----> {650}:
delay-after-clid-before-ring: ----> {250}:
delay-before-dtas-after-lr: -----> {250}:
vmwi-mode: -----> {dtasets}:
delay-before-vmwi-after-dtas: ----> {50}:
delay-before-vmwi-after-rpas: ----> {650}:
vmwi-delay-before-dtas-after-lr: -> {250}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Changing the jitter buffer

The type and size of the jitter buffer in the MXK can be configured. The jitter buffer accommodates the packets received, so that the inter-arrival jitter of the packets received does not degrade the voice quality. Without a jitter buffer, some inter-arrival jitter changes would be late, which would have the same effect as lost packets. The jitter buffer also reorders the out-of-order packets received.

Modify the following parameters in the **voice-dsp-default-profile** to change jitter buffer:

Table 39: Configurable jitter buffer parameters

Parameter	Description
jitter-buffer-type	There are two types of jitter algorithms: static and dynamic. Values: static A static jitter buffer does not change to compensate for inter-arrival jitter changes. Default jitter buffer type is static for VoATM applications. dynamic Allows the jitter buffer to grow and shrink as inter-arrival jitter changes. Default jitter buffer type is dynamic for VoIP applications.
jitter-buffer-size	Specifies the size of the jitter buffer. Values: 1 to 160 Note that changes to the jitter buffer are based on 5 ms frame sizes. For example: 1 to 5 = 5 ms 6 to 10 = 10 ms 11 to 15 = 15 ms 16 to 20 = 20 ms ... 146 to 150 = 150 ms 151 to 155 = 155 ms 156 to 160 = 160 ms Default: 10



Note: Any changes made to jitter buffer size and jitter buffer type take effect in the next call.

To change the type and size of the jitter buffer:

```

zSH> update voice-dsp-default-profile 0
Please provide the following: [q]uit.
redundancy-over-subscription-type: -> {high}:
jitter-buffer-type: -----> {dynamic}: static
jitter-buffer-size:-----> {10}: 22
inter-arriv-jit-threshold: -----> {80}:
pkts-lost-threshold: -----> {600}:
echo-cancellation-type: -----> {g165echotl48}:
silence-supression-type: -----> {silsupoff}:
echo-return-loss: -----> {erl0db}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Configuring signal type and ring frequency

Modify the following parameters in the **analog-fxs-cfg-profile** if you need to change signalling type and ring frequency for each voice line:

Table 40: Configurable signalling type and ring frequency parameters

Parameter	Description
signal-type	The method by which an off-hook condition is indicated. Values: fxsloopstart fxsgroundstart Default: fxsloopstart
ring-frequency	Rate in cycles per second (Hertz) at which polarity reversal occurs on ringing. Values: ringfrequency20 ringfrequency25 ringfrequency30 ringfrequency50 Default: ringfrequency20
ring-back	The ring back is requested if this variable is set to on. Values: on off Default: off

If you need to modify the signaling and ring frequency, update the **analog-fxs-cfg-profile** for each interface. For example:

```
zSH> update analog-fxs-cfg-profile 1-3-1-0/voicefxs
signal-type: ----> {fxsloopstart} fxsgroundstart
ring-frequency: -> {ringfrequency20}
ring-back: -----> {off}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Configure an IP interface for voice traffic

Configure a network facing IP interface and route for voice traffic when configuring the MXK for any of the voice signaling protocols.

Configuring the IP voice path

Create a network facing IP interface that will pass the voice traffic to the network.

- 1 Create an IP interface for VoIP, in this case on the network facing Ethernet port, and designate a VLAN, CoS and ToS values.

Note that the IP interface cannot be created on a management port (i.e. 1-a-1-0).

```
zSH> interface add 1-a-2-0/eth vlan 100 192.168.127.104/24
Created ip-interface-record ethernet2-100/ip.
```



Note: IPv4 is supported for IP termination on the MXK. IPv6 is not supported for IP termination on the MXK.

Verify the IP interface.

```
zSH> interface show
2 interfaces
Interface      Status  Rd/Address          Media/Dest Address  IfName
-----
1/a/1/0/ip     UP      1 172.16.160.49/24  00:01:47:1a:fe:64   ethernet1
1/a/2/0/ip     UP      1 192.168.127.104/24 00:01:47:1a:fe:66   ethernet2-100
-----
```

- 2 Add a specific route for the VoIP server or VoiceGateway MALC.

```
zSH> route add 10.10.10.0 255.255.255.0 192.168.127.254
```

Verify the route.

```
zSH> route list
Domain  Dest          Mask          Nexthop          IfNum  Cost  Enable
-----
1       10.10.10.0    255.255.255.0 192.168.127.254  0      1    enabled
```

- 3 Or add a default route for the VoIP server or VoiceGateway MALC.

```
zSH> route add default 192.168.127.254 1
```

Verify the route.

```
zSH> route list
Domain  Dest          Mask          Nexthop          IfNum  Cost  Enable
-----
1       0.0.0.0       0.0.0.0       192.168.127.254  0      1    enabled
```

Voice add command



Caution: Don't delete the **ip-interface-record** profile after creating a voice connection on that interface.



Note: You can use the **voice status** and/or **voice ring** command to verify a POTS voice connection. Note that the **voice ring** command will ring the subscriber's phone.

Before creating VoIP connections, make sure the IP interface for voice and VoIP server settings are properly configured.

POTS subscribers are connected to VoIP remote endpoints by the **voice add** command.

voice add *subscriber-endpoint remote-endpoint*

- The following VoIP *subscriber-endpoint* parameter and options are available:

pots interface [**alaw**|**mulaw**]

Select a-law or mu-law for the subscriber only if necessary. The default value depends on which country specified in the **countryregion** parameter of the **system** profile.

isdn interface [**alaw**|**mulaw**]

Set ISDN to VoIP connection. For details refer to [ISDN to VoIP connection with SIP PLAR, page 505](#) and [ISDN to VoIP connection with H.248, page 513](#).

- The following VoIP *remote-endpoint* parameters and options are available:

voip *IpIfname dn dir-num* [**name** *username*] [**pw** *password*] [**plar** *dest-ipaddr*] [**reg** *serverId*] [**codec** *pref-codec*][**t38fax** *t38-fax*]

By default, the **reg serverId** is set to 1. It means MXK uses the primary VoIP server that is specified in the **voip-server-entry** 1/x (any *addrIndex*) profile. The *serverId* is refer to the *serverId* in the **voip-server-entry** *serverId/addrIndex* profile. There is a special case for SIP PLAR in which the default value of **reg serverId** is 0, and the information of this SIP PLAR server is in the **voip-server-entry** 255/255.

Supported **codecs** are:

- g711mu (the default setting if the country code is set to mu-law)
- g711a (the default setting if the country code is set to a-law)
- g729a

The MXK G.729A VoIP compression provides an optional fallback mode to G.711. The parameter for the fallback mode is **g711-fallback** and is set in the **subscriber-voice-voip** profile. The default settings for the **subscriber-voice-voip** profile are:

- preferred-codec: g711mu (if the **countryregion** uses mu-law) or g711a (if the **countryregion** uses a-law)
- g711-fallback: true (relevant with g729a)
- frames-per-packet: 4
- t38-fax: t38none
- hotline-initial-timer: 4



Note: For MGCP and H.248 calls, the MXK always use the codec provided by the MGCP server or media gateway controller. If the MGCP server or media gateway controller didn't provide the codec, then the MXK uses the preferred-codec settings.

SIP

- [SIP server on page 483](#)
- [SIP dial plan configuration on page 485](#)
- [POTS to VoIP connection with SIP on page 487](#)
- [Emergency Stand Alone \(ESA\) for SIP on page 489](#)
- [DSCP marking for SIP and RTP on page 493](#)
- [Enhanced SIP 911 Service on page 495](#)
- [RFC 3262 for SIP on page 497](#)

SIP server



Note: Redundant SIP server support is implemented through DNS lookups for only BroadSoft Broadworks switch configurations.

SIP signaling identifies callers and callees by SIP addresses and allows signals to be redirected to proxy servers.

The MXK supports single softswitch configurations for SIP.



Note: If all SIP subscribers do not register after a system reboot, increase the **server-max-timer** value in the **voice-system** profile to a higher value, for example 180 seconds. The default value is 20 seconds.

Configuring a SIP server

To configure SIP:

- 1 Create the **voip-server-entry** profiles to specify the VoIP server groups and IDs.

Specify the **voip-server-entry** profile with server ID and address index numbers. This example configures a SIP server in server ID *1* with address index *1*.

This example keeps the default value 1 in the **message-retry-count** field. This field is used when SIP register (in voip-server-entry) or outbound proxy sever (in sip-dialplan profile) are configured as DNS name. MxK tries to do a srv lookup of the DNS name and caches the primary & secondary IP addresses. This field specifies the number of retries of SIP message to every DNS resolved server IP addresses. By default the SIP message will be retransmitted to the first DNS resolved IP address once and the remaining retransmissions will be to the second DNS resolved IP address. If the DNS name is resolved to a single IP address all retransmissions will be to the single IP address. In the range of 1 to 10.

```

zSH> new voip-server-entry 1/1
voip-server-entry 1/1
Please provide the following: [q]uit.
zhoneVoipServerAddrType: -----> {ipv4}:
zhoneVoipServerAddr: -----> {}: 192.168.49.1
zhoneVoipServerUdpPortNumber: -----> {5060}:
zhoneVoipServerId: -----> {generic}: metaswitch
protocol: -----> {sip}:
sendCallProceedingTone: -----> {false}:
rtcpEnabled: -----> {false}:
rtcpPacketInterval: -----> {5000}:
interdigitTimeOut: -----> {10}:
ipTos: -----> {0}:
systemDomainName: -----> {}:
expires-invite-value: -----> {3600}:
expires-register-value: -----> {3600}:
expires-header-method: -----> {register}:
session-timer: -----> {off}:
session-expiration: -----> {180}:
session-min-session-expiration: ---> {180}:
session-caller-request-timer: -----> {no}:
session-callee-request-timer: -----> {no}:
session-caller-specify-refresher: -> {omit}:
session-callee-specify-refresher: -> {uac}:
dtmf-mode: -----> {rfc2833}:
rtp-termid-syntax: -----> {}:
rtpDSCP:-----> (0)
signalingDSCP:-----> (0)
dtmf-payload-id: -----> {101}:
register-ready-timeout: -----> {10}:
message-retry-count: -----> {1}:
voip-features: -----> {NONE(0)}:
transport-protocol: -----> {udp}:
signalling-local-port-number: -----> {5060}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```



Note: IPv4 is supported for IP termination on the MXK. IPv6 is not supported for IP termination on the MXK.

The **zhoneVoipServerAddr** parameter will only accept IPv4 addresses.

2 Create a SIP dialplan for the SIP server.

In each dialplan, specify the desired call parameters and use the **voip-server-entry** parameter to identify the server ID for which the dialplan is used. This example references server ID 1. See [SIP dial plan configuration on page 485](#) for more information.

```
zSH> new sip-dialplan 0
```



```

sip-dialplan 0
Please provide the following: [q]uit.
match-string: -----> {}: *x.T | x.T
sip-ip-address: -----> {0.0.0.0}: 192.168.49.1
destination-name: -----> {}:
number-of-digits: -----> {0}:
prefix-strip: -----> {0}:
prefix-add: -----> {}:
dialplan-type: -----> {normal}:
voip-server-entry-index: -----> {0}: 1
override-interdigit-timeout: -> {0}: 3
dialplan-class: -----> {NONE(0)}:
description: -----> {}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```

SIP dial plan configuration

A dialing plan for POTS-to-SIP outgoing calls consists of a series of acceptable dial strings and the corresponding IP addresses to which SIP control messages are sent to initiate the call.

Each dial string is represented as digits, wildcards, and regular-expression-like patterns according to the following rules:

- Digits 0 to 9 are allowed as well as * and #.
- The character **x** to indicate a wildcard for 0 or more digits between 0-9.
- A dial-string character **T** can be used in the **override-interdigit-timeout** parameter value in the SIP dialplan.

Examples:

- **0T** for the number zero and nothing else.
- **011T** for numbers 011 then any number of digits before the interdigit time out.
- **9T** for the number 9 and any number of digits before the interdigit time out.
- **#T** anything followed by a # and an interdigit time out.
- A digit range can be specified using brackets [], as follows:
[135] means digits 1, 3, or 5.
[1-4] means digits 1, 2, 3, or 4.
- MGCP-style digit mapping where a period ‘.’ represents any digit and a | character indicates an inclusive OR.

Examples:

- **.T** for any number of digits before the interdigit timeout.

- ***x.T | x.T** indicates star plus any number of digits followed by the inter-digit timeout or any number of digits followed by the inter-digit timeout.
- ***.xT | x.T | [2-9]11** indicates star plus any number of digits followed by the inter-digit timeout or any number of digits followed by the inter-digit timeout, or digits 2 to 9 followed by 11. The [2-9]11 explicit digit matching enables expedited call connections for emergency calls.

Table 41 describes the configurable **sip-dialplan** profile parameters for outgoing VoIP calls.

Table 41: sip-dialplan profile parameters

Parameter	Description
match-string	A dial string against which collected digits are matched.
sip-ip-address	Upon detecting a match between the collected digits and the dial string, this IP address is used for SIP negotiations to initiate the call.
destination -name	User-specified name of the destination for the dial string.
number-of-digits	Number of digits to wait for before initiating the call.
prefix-strip	Number of prefix digits to strip from dialled digits.
prefix-add	String to be added to the beginning of the dialled digits before call initiation.
dialplan-type	Type of the dial plan. Dialplan types are: • normal • callpark • esa • isdnsig • intercom
voip-server-entry-index	An index to associated voip-server-entry for this sip-dialplan . This index references the registration server specified in the voip-server-entry profile.
override-interdigit-timeout	Override the partial-dial-timeout value in voice-system profile.

Table 41: sip-dialplan profile parameters (Continued)

Parameter	Description
dialplan-class	Use this field to enable or disable the Enhanced SIP 911 service. For the details, refer to Enhanced SIP 911 Service, page 495. When the Enhanced SIP 911 service has been enabled, there will be indication on the MXK that if there is a emergency call (e.g. 911) in progress, the system will not allow itself to be upgraded/rebooted. Values: NONE To disable the Enhanced SIP 911 service, specify None in the dialplan-class field. This is the default value. emergency To enable the Enhanced SIP 911 service, specify emergency in the dialplan-class field and specify the match-string field as regular expression for emergency call number or emergency call number itself (e.g 911).
description	Brief description about the sip-dialplan.

```

zSH> new sip-dialplan 1
Please provide the following: [q]uit.
match-string: -----> {}: 510555101[1-9]
sip-ip-address: -----> {0.0.0.0}: 192.168.88.199
destination-name: -----> {}:
number-of-digits: -----> {0}: 10
prefix-strip: -----> {0}: 1
prefix-add: -----> {}: 0
dialplan-type: -----> {normal}:
voip-server-entry-index: -----> {0}: 1
override-interdigit-timeout:--> {0}: 22
dialplan-class: -----> {NONE(0)}:
description: -----> {}:west campus
.....
Save new record? [s]ave, [c]hange or [q]uit: s
Record created.

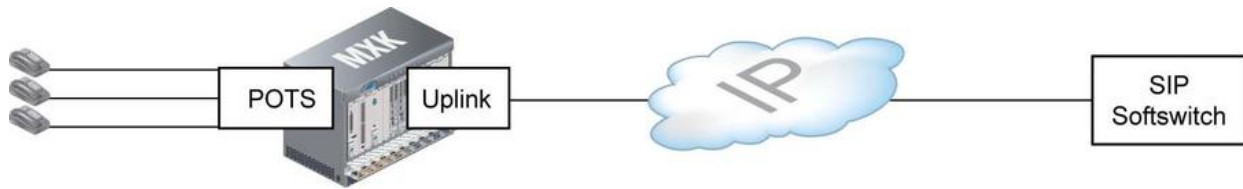
```

POTS to VoIP connection with SIP

After configured system settings, IP interface, and SIP server settings properly, user can create POTS to SIP softswtich connections. And note that MXK only support one VoIP signaling protocol at a time.

The following figure shows for POTS to SIP softswtich configuration, the MXK interconnects POTS terminal equipment directly to SIP softswitches.

Figure 63: MXK common voice configuration - POTS to SIP Softswitch



Creating POTS to SIP softswitch connections

This example creates a POTS to SIP softswitch connection:

- 1 Verify/create an IP interface for voice traffic
See [Configure an IP interface for voice traffic on page 480](#).
- 2 Verify/create the SIP server.
See [Configuring a SIP server on page 483](#).
- 3 Use the **voice add** command to add the POTS to VoIP connection. This example creates a connection with a directory number **201202999** and the name **201202999**. The VoIP remote-endpoint *user name* is case sensitive and must match the voice switch requirements, the following example is for SIP, the name matches the directory number.

```
zSH> voice add pots 1-10-1-0/voicefxs voip ethernet2-100/ip dn 201202999
name 201202999 pw password
Created subscriber 1/5
Created subscriber-voice 1/5/1
Created subscriber-voice-pots 1
Created subscriber-voice-voip 2
```

This example didn't specify the **reg** option, it means the MXK uses the primary VoIP server that is specified in the **voip-server-entry 1/x** (any address index) profile.

- 4 View the voice connection.

```
zSH> voice show
```

Subscriber end-point	Remote End point	Username	SRV STA
1-10-1-0/voicefxs	ethernet2-100/ip	201202999	1 ENA

Total number of voice connections : 1

- 5 The **voice ring** command can be used to verify a POTS voice connection without placing a call. The **voice status** command can be used to display runtime voice port status, verify the phone's ring status if the ringing cannot be heard, and display interface group status.

Emergency Stand Alone (ESA) for SIP

This section describes ESA SIP support on the MXK:

- [Configuring VoIP ESA clusters, page 489](#)
- [Configuring ESA for 911 calls, page 492](#)
- [Verifying ESA, page 492](#)

For VoIP SIP voice connections, the MXK provides emergency calling services during network or equipment failures that cause a loss of connection to the configured SIP server or voice gateway MALC.

For VoIP SIP connections, the ESA feature enables numbers configured within ESA dialplans to communicate with any residences or businesses specified as the destination of the dialplans in an ESA cluster of MXK devices. Incoming calls from outside the ESA group and outgoing calls to numbers outside the ESA cluster receive a fast-busy signal.

When ESA is activated, call features such as call waiting, are not supported.

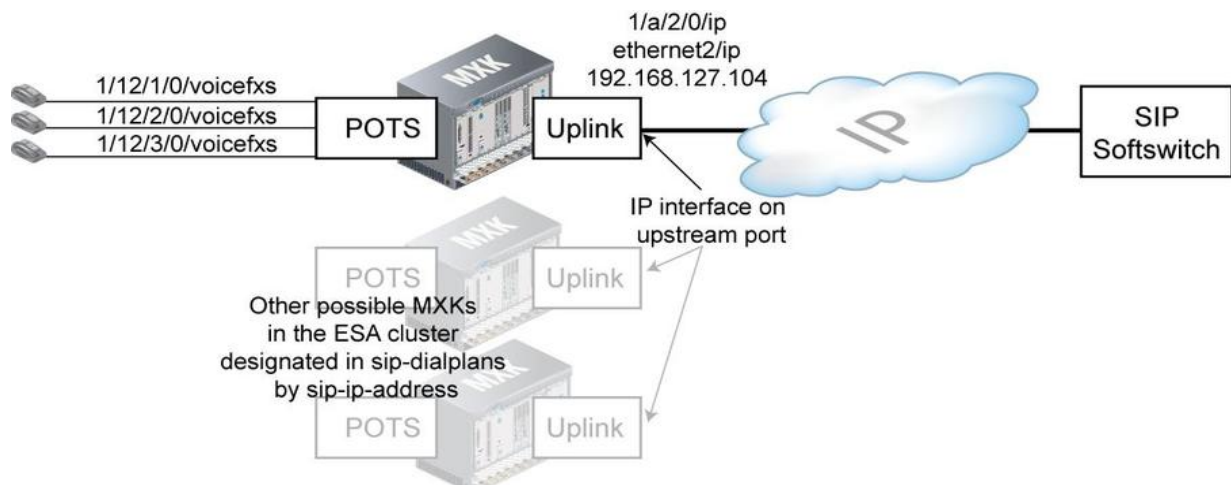


Note: After a loss of connection to the SIP server, there may be a delay up to 5 minutes before ESA notification is received and ESA features are accessible.

There may be a similar delay before resuming normal calling after the outage is restored.

Figure 64 illustrates ESA support for VoIP SIP connections.

Figure 64: ESA for VoIP SIP connections



Configuring VoIP ESA clusters

VoIP ESA cluster requires an ESA SIP dialplan in each of the SLMS device that participate in the ESA cluster. MXK For each ESA dialplan, enter the IP addresses of the desired MXK in the **sip-ip-address** field and change the

dialplan-type to **esa**. Also, if desired, change the **destination-name** to the target MXK.

When in ESA mode, the MXK sequentially checks the configured dialplans for a matching string starting with the lowest number to the highest number dialplan. If a match is found, the call connection process is initiated immediately. If a match is not found, the next sequential dialplan is checked until all configured dialplans have been checked. Calls with unmatched strings are then terminated. It is recommended to configure lower number dialplans for more frequently called nodes and higher number dialplans for less frequently called nodes.

This example creates VoIP server 1/1 and creates SIP dialplan 1 for the VoIP server. SIP dialplan 2 is used on MXK 1 with IP address 172.24.94.219; SIP dialplan 3 is used on MXK 2 with IP address 172.24.94.222. SIP dialplan 4 is used on MXK 3 with IP address 172.24.94.223. It also sets the match-string to '*x.T | x.T' to accept all numbers, all number of digits, and the dialplan type to ESA. This dialplan enables ESA calls to connect to other subscribers within the same MXK. Additional dialplans are created for each of the neighboring MXK nodes.



Note: A SIP dialplan of type normal should be configured and connected to a VoIP SIP server for non-ESA calls.

- 1 Configure a SIP server in server ID 1 with address 1. The IP address of this SIP server is 172.16.60.1.

```
zSH> new voip-server-entry 1/1
Please provide the following: [q]uit.
zhoneVoipServerAddrType: -----> {ipv4}:
zhoneVoipServerAddr: -----> {}: 172.16.60.1
zhoneVoipServerUdpPortNumber: -----> {5060}:
zhoneVoipServerId: -----> {generic}:metaswitch
protocol: -----> {sip}:
sendCallProceedingTone: -----> {false}:
rtcpEnabled: -----> {false}:
rtcpPacketInterval: -----> {5000}:
interdigitTimeout: -----> {10}:
ipTos: -----> {0}:
systemDomainName: -----> {}:
expires-invite-value: -----> {3600}
expires-register-value: -----> {3600}
expires-header-method: -----> {register}
session-timer: -----> {off}
session-expiration: -----> {180}
session-min-session-expiration: ---> {180}
session-caller-request-timer: -----> {no}
session-callee-request-timer: -----> {no}
session-caller-specify-refresher: -> {omit}
session-callee-specify-refresher: -> {uac}
dtmf-mode:-----> (rfc2833)
rtp-termid-syntax:-----> ()
rtpDSCP:-----> (0)
```

```
signalingDSCP:-----> (0)
.....
Save new record? [s]ave, [c]hange or [q]uit: s
Record created.
```

2 Create a SIP dialplan for SIP server. Refer to server ID 1.

```
zSH> new sip-dialplan 1
match-string: -----> {0} *x.T | x.T
sip-ip-address: -----> {0} 172.16.60.1
destination-name: -----> {}
number-of-digits: -----> {0}
prefix-strip: -----> {0}
prefix-add: -----> {}
dialplan-type: -----> {normal}
voip-server-entry-index: -----> {0} 1
override-interdigit-timeout: -> {0}
dialplan-class: -----> {NONE(0)}:
description: -----> {}: dialplan1forserver1
```

3 Create a SIP dialplan for MXK #1:

```
zSH> new sip-dialplan 2
match-string: -----> {0} *x.T | x.T
sip-ip-address: -----> {0} 172.24.94.219
destination-name: -----> {}
number-of-digits: -----> {0}
prefix-strip: -----> {0}
prefix-add: -----> {}
dialplan-type: -----> {normal}esa
voip-server-entry-index: -----> {0}
override-interdigit-timeout: -> {0}

dialplan-class: -----> {NONE(0)}:
description: -----> {}: dialplan2forMXK1
```

4 Create additional SIP dialplans for so ESA calls can connect to subscribers on other SLMS devices. This dialplan allows ESA calls to connect to subscribers on MXK #2.

```
zSH> new sip-dialplan 3
match-string: -----> {0} *x.T | x.T
sip-ip-address: -----> {0} 172.24.94.222
destination-name: -----> {}
number-of-digits: -----> {0}
prefix-strip: -----> {0}
prefix-add: -----> {}
dialplan-type: -----> {normal}esa
voip-server-entry-index: -----> {0}
override-interdigit-timeout: -> {0}
dialplan-class: -----> {NONE(0)}:
description: -----> {}: dialplan3forMXK2
```

5 This dialplan allows ESA calls to connect to subscribers on MXK #3.

```

zSH> new sip-dialplan 4
match-string: -----> {0} *x.T | x.T
sip-ip-address: -----> {0} 172.24.94.223
destination-name: -----> {}
number-of-digits: -----> {0}
prefix-strip: -----> {0}
prefix-add: -----> {}
dialplan-type: -----> {normal}esa
voip-server-entry-index: -----> {0}
override-interdigit-timeout: -> {0}
dialplan-class: -----> {NONE(0)}:
description: -----> {}:dialplan4forMXK3

```

Configuring ESA for 911 calls

To configure ESA for VoIP connections for 911 calls, create an ESA dialplan with a match-string of 911 and the IP address of the MXK shelf in the **sip-ip-address** field. Also, change the prefix-strip to 3. The prefix-strip setting deletes the dialed 911 numbers. Enter the desired phone number to be called in the prefix-add field. This number must be a valid voicefxs line in the same MXK shelf. Change the dial-plan type to **esa**.

This example creates a SIP dialplan called 911 on the MXK with IP address 172.24.94.219. It replaces the dialed 911 number with the phone number 7281001 and changes the dialplan type to ESA.

```

zSH> new sip-dialplan 911
match-string: -----> {}911
sip-ip-address: -----> {0} 172.24.94.219
destination-name: -----> {}
number-of-digits: -----> {0}
prefix-strip: -----> {0}3
prefix-add: -----> {}7281001
dialplan-type: -----> {normal}esa
voip-server-entry-index: -----> {0}
override-interdigit-timeout: -> {0}
dialplan-class: -----> {NONE(0)}:
description: -----> {}:

```

Verifying ESA

Verify whether ESA support is in-use.

- 1 Enter the **voice status** command. This command lists the voice port, destination, call state, and ESA state along with other status information

```

zSH> voice status
port      term state  destination          call state  hook  ring  ESA
----      -
1-12-1-0/voicefxs  UP        VoIP:69:VoIP EndPtIdx-152  No call  ON    NoRing
ON

```



```
1-12-2-0/voicefxs UP VoIP:69:VoIP EndPtIdx-154 No call ON NoRing
ON
```

2 Or you can use the **sipstack esa** command.

```
zSH> sipstack esa
sip server: 172.16.60.1:5060, Dns: 172.24.94.2 status: Not resolved # of
sub: 72 , esaMode(ip): ON
```

DSCP marking for SIP and RTP

The VOIP traffic has two parts: signalling and RTP (Real-Time Transport Protocol) traffic. SIP-based telephones use SIP (Session Initiation Protocol) for the call setup, and RTP for transport of the audio packets.

Instead of using COS to DSCP mapping on other devices (such as ONTs or telephones), users now can prioritize traffic in the network by marking SIP signalling packets and RTP packets with different DSCP (Differentiated Services Code Point) values on the MXK. When the SIP or RTP packets originate from the MXK, they have different priorities according to what DSCP values are configured by users. Note that the MXK only marks the packets, it does not perform any actions based on DSCP values.

The value range of the DSCP values is from 0 to 63. 0 is the default value, it means none DSCP values are marked. Those values are in decimal format, or the PHB Classes. The table below lists some common DSCP values in decimal format and their matching PHB classes. You can enter the DSCP values either in decimal format or in PHB class format.

Table 42: Mapping between DSCP values in decimal and DSCP/PHB classes

DSCP values in Decimal format	DSCP/PHB Class	DSCP values in Decimal format	DSCP/PHB Classes
0	none	28	af32
8	cs1	30	af33
10	af11	32	cs4
12	af12	34	af41
14	af13	36	af42
16	cs2	38	af43
18	af21	40	cs5
20	af22	46	ef
22	af23	48	cs6
24	cs3	56	cs7
26	af31		

Configuring DSCP marking for SIP and RTP

To add or modify DSCP markings for SIP packets and RTP packets on the MXK, use the **new voip-server-entry** or **update voip-server-entry** command.

- 1 Specify the desired values for the VoIP server, such as Server Address and Server Id, etc.

To add DSCP marking for SIP packets, enter a value to the **signalingDSCP** field.

To add DSCP marking for RTP packets, enter a value to the **rtpDSCP** field.

```

zSH> new voip-server-entry 1/1
Please provide the following: [q]uit.
zhoneVoipServerAddrType: -----> {ipv4}:
zhoneVoipServerAddr: -----> {}: 172.16.60.1
zhoneVoipServerUdpPortNumber: -----> {5060}:
zhoneVoipServerId: -----> {generic}:metaswitch
protocol: -----> {sip}:
sendCallProceedingTone: -----> {false}:
rtcpEnabled: -----> {false}:
rtcpPacketInterval: -----> {5000}:
interdigitTimeout: -----> {10}:
ipTos: -----> {0}:
systemDomainName: -----> {}:
expires-invite-value: -----> {3600}
expires-register-value: -----> {3600}
expires-header-method: -----> {register}
session-timer: -----> {off}
session-expiration: -----> {180}
session-min-session-expiration: ---> {180}
session-caller-request-timer: -----> {no}
session-callee-request-timer: -----> {no}
session-caller-specify-refresher: -> {omit}
session-callee-specify-refresher: -> {uac}
dtmf-mode:-----> (rfc2833)
rtp-termid-syntax:-----> ()
rtpDSCP:-----> (0) 1
signalingDSCP:-----> (0) cs1                                cs1maps to 8
.....
Save new record? [s]ave, [c]hange or [q]uit: s
Record created.

```

- 2 To modify the DSCP values, use the **update voip-server-entry** command.

```

zSH> update voip-server-entry 1/1
voip-server-entry 1/1
Please provide the following: [q]uit.
zhoneVoipServerAddrType: -----> {ipv4}:
zhoneVoipServerAddr: -----> {172.16.60.1}:
zhoneVoipServerUdpPortNumber: -----> {5060}:

```

```

zhoneVoipServerId: -----> {metaswitch}:
protocol: -----> {sip}: ** read-only **
sendCallProceedingTone: -----> {false}:
rtcpEnabled: -----> {false}:
rtcpPacketInterval: -----> {5000}:
interdigitTimeOut: -----> {10}:
ipTos: -----> {0}:
systemDomainName: -----> {}:
expires-invite-value: -----> {3600}:
expires-register-value: -----> {3600}:
expires-header-method: -----> {register}:
session-timer: -----> {off}:
session-expiration: -----> {180}:
session-min-session-expiration: ---> {180}:
session-caller-request-timer: -----> {no}:
session-callee-request-timer: -----> {no}:
session-caller-specify-refresher: -> {omit}:
session-callee-specify-refresher: -> {uac}:
dtmf-mode: -----> {rfc2833}:
rtp-termid-syntax: -----> {}:
rtpDSCP: -----> {1}: 9
signalingDSCP: -----> {cs1}: 7
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Enhanced SIP 911 Service



Note: Enhanced SIP 911 service is supported on all the POTS line cards, and supported on the MXK-POTS-EBS-PKT-24 card in ESA mode.

With the enhanced SIP 911 service in the MXK system, if there is a emergency call (e.g. 911) in progress, there will be indications on the MXK, and the system will not allow itself to be upgraded or rebooted unless users force to do so.

Note that the emergency numbers differ from country to country. This section uses 911 as an example.

Certain user operations will cause the system to be upgraded or rebooted, thus will be prevented to use, such as **systemreboot**, **slotreboot**, **voice down**, **voice bounce**, **voice delete**, **port down**, **port bounce**, **voip-server rereg**, **swupgrade**, **swact**, and **upgrade**.

If any of the above commands are issued while emergency calls are in progress, the MXK will inform users there are active emergency calls in progress and exit the operation. If users still want to continue with requested operation even when emergency call(s) exists, they must use the command with “**-force**” option to execute the command forcibly. For example:

```
zSH> systemreboot
```

```
Action denied due to active emergency call(s); Use -force
option to override!!
```

Use force option to perform the action even when there is an active emergency call(s) in the system.

```
zSH> systemreboot -force
```

Active emergency call(s) exists, do you still want to continue [yes] or [no]: **yes**

Do you want to reboot the system? (yes or no) [no] **yes**



Note: It is not recommended to force reboot or shutdown system while there is emergency call in progress. Only do it when it is necessary.

With the enhanced SIP 911 feature in the MXK system, once the emergency call is active, this emergency call should be disconnected by the emergency call operator only. The detail emergency call process listed as below:

1. The caller picks up the phone.
2. The caller dials emergency call number 911.
3. The caller hangs up the phone.
4. The emergency call is still active.
 - If the caller picks up the phone, he will hear dialtone. When the caller starts to dial, after the third digit, the call will be connected back to the 911 operator.
 - If the caller does not pick the phone back up, the 911 operator still be able to force a ring on the phone.
5. The 911 operator terminates the 911 call.

Enabling the Enhanced SIP 911 service

To enable the enhanced SIP 911 feature, create another sip-dialplan profile with the match-string as regular expression for the emergency call number or the emergency call number (e.g. 911) and the dialplan-class as emergency.

Created a new sip-dialplan for emergency calls.

```
zSH> new sip-dialplan 4
sip-dialplan 4
Please provide the following: [q]uit.
match-string: -----> {}: 911
sip-ip-address: -----> {0.0.0.0}:
destination-name: -----> {}: proxy.example.com
number-of-digits: -----> {0}:
prefix-strip: -----> {0}:
prefix-add: -----> {}:
dialplan-type: -----> {normal}:
voip-server-entry-index: -----> {0}: 1
override-interdigit-timeout: -> {0}: 3
dialplan-class: -----> {NONE(0)}: emergency
description: -----> {}: E911
.....
```

```
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Displaying the emergency calls in the system

Use the **voicestat** command to show whether there are emergency calls in the system, and how many emergency calls in the system.

- 1 Display the system level voice stats.

```
zSH> voicestat system
***** System Voice Stats *****
Incoming blocked           0
Incoming completed        2
Outgoing blocked           0
Outgoing completed        1
Active calls               2
Emergency calls            1
ESA calls                  0
```

- 2 Display the card level voice stats. The first 1 is the shelf ID, the second 1 is the slot ID.

```
zSH> voicestat card 1 1
***** Card Voice Stats *****
Incoming blocked           0
Incoming completed        2
Outgoing blocked           0
Outgoing completed        1
Active calls               2
Emergency calls            1
ESA calls                  0
```

- 3 Display the subscriber level voice stats. 1 is the subscriber endpoint index.

```
zSH> voicestat subscriber 1
***** Subscriber Voice Stats *****
Incoming blocked           0
Incoming completed        0
Outgoing blocked           0
Outgoing completed        1
Active calls               1
Is in Emergency call       Yes
Is in ESA mode             No
```

RFC 3262 for SIP

RFC 3262 (Reliability of Provisional Responses in the SIP) feature is implemented in MXK. This is an extension to Session Initiation Protocol (SIP) providing reliable provisional response messages. This extension uses the option tag 100rel and defines the Provisional Response

ACKnowledgement (PRACK) method. RFC 3262 is achieved by mirroring reliability mechanism for 2xx final responses to INVITE. Reliable provisional responses are retransmitted with an exponential backoff. These retransmissions cease when a PRACK message is received. PRACK messages plays the same role as ACK for 2xx final response for INVITE message. However, like BYE, unlike ACK, PRACK has its own response.

When RFC 3262 is enabled on MXK, the far end also has to support RFC 3262 to achieve reliability of provisional messages. Otherwise, SIP signaling will fall back to provisional messages without reliability (RFC 3261).

Enabling RFC 3262 for SIP

By default, RFC 3262 is disabled irrespective of the value of voip-features in voip-server. To add or modify RFC 3262 settings for SIP packets on the MXK, use the **new voip-server-entry** or **update voip-server-entry** command.

1 Check the current settings.

```
zSH> get voip-server-entry 1/1
voip-server-entry 1/1
zhoneVoipServerAddrType: -----> {dns}
zhoneVoipServerAddr: -----> {metaswitch.oak.zhone.com}
zhoneVoipServerUdpPortNumber: -----> {5060}
zhoneVoipServerId: -----> {generic}
protocol: -----> {sip}
sendCallProceedingTone: -----> {false}
rtcpEnabled: -----> {false}
rtcpPacketInterval: -----> {5000}
interdigitTimeOut: -----> {10}
ipTos: -----> {0}
systemDomainName: -----> {metaswitch.oak.zhone.com}
expires-invite-value: -----> {3600}
expires-register-value: -----> {3600}
expires-header-method: -----> {register}
session-timer: -----> {off}
session-expiration: -----> {180}
session-min-session-expiration: ---> {180}
session-caller-request-timer: -----> {no}
session-callee-request-timer: -----> {no}
session-caller-specify-refresher: -> {omit}
session-callee-specify-refresher: -> {uac}
dtmf-mode: -----> {rfc2833}
rtp-termid-syntax: -----> {}
rtpDSCP: -----> {0}
signalingDSCP: -----> {0}
dtmf-payload-id: -----> {101}
register-ready-timeout: -----> {10}
message-retry-count: -----> {1}
voip-features: -----> {NONE(0)}
transport-protocol: -----> {udp}
signalling-local-port-number: -----> {5060}
```

- 2** To kick start this feature, the config bit “Rfc3262” has to be set for the voipserver entry. Note that this field is case-sensitive.

```
zSH> voipserver 1/1 set bits +Rfc3262
```

```
zSH> voipserver 1/1 show bits
```

```
Behavior String: +Rfc3262
```

- 3** To enable rfc 3262 on the SIP packets:

```
zSH> update voip-server-entry 1/1
```

```
voip-server-entry 1/1
```

```
Please provide the following: [q]uit.
```

```
zhoneVoipServerAddrType: -----> {dns}:
zhoneVoipServerAddr: -----> {metaswitch.oak.zhone.com}:
zhoneVoipServerUdpPortNumber: -----> {5060}:
zhoneVoipServerId: -----> {generic}:
protocol: -----> {sip}: ** read-only **
sendCallProceedingTone: -----> {false}:
rtcpEnabled: -----> {false}:
rtcpPacketInterval: -----> {5000}:
interdigitTimeout: -----> {10}:
ipTos: -----> {0}:
systemDomainName: -----> {metaswitch.oak.zhone.com}:
expires-invite-value: -----> {3600}:
expires-register-value: -----> {3600}:
expires-header-method: -----> {register}:
session-timer: -----> {off}:
session-expiration: -----> {180}:
session-min-session-expiration: ---> {180}:
session-caller-request-timer: -----> {no}:
session-callee-request-timer: -----> {no}:
session-caller-specify-refresher: -> {omit}:
session-callee-specify-refresher: -> {uac}:
dtmf-mode: -----> {rfc2833}:
rtp-termid-syntax: -----> {}:
rtpDSCP: -----> {0}:
signalingDSCP: -----> {0}:
dtmf-payload-id: -----> {101}:
register-ready-timeout: -----> {10}:
message-retry-count: -----> {1}:
voip-features: -----> {NONE(0)}: rfc3262
transport-protocol: -----> {udp}:
signalling-local-port-number: -----> {5060}:
.....
```

```
Save changes? [s]ave, [c]hange or [q]uit: s
```

```
Record updated.
```

SIP PLAR

- [SIP PLAR server configuration on page 500](#)
- [ESA for SIP PLAR on page 501](#)
- [POTS to VoIP connection with SIP PLAR on page 504](#)
- [ISDN to VoIP connection with SIP PLAR on page 505](#)

SIP PLAR server configuration

User do not need to create a SIP PLAR server entry, the SIP PLAR server is automatically created when user specifying the **voice add** command with the **plar** option.

Viewing a SIP PLAR server

This entry serves as the default server entry.

Use the **get voip-server-entry** *serverID/addrIndex* command to view the SIP PLAR server entry. The *serverID/IndexID* must be 255/255. The **zhoneVoipServerAddr** must be 0.0.0.0.

```
zSH> get voip-server-entry 255/255
voip-server-entry 255/255
Please provide the following: [q]uit.
zhoneVoipServerAddrType: -----> {ipv4}:
zhoneVoipServerAddr: -----> {0.0.0.0}:
zhoneVoipServerUdpPortNumber: -----> {5060}:
zhoneVoipServerId: -----> {generic}:
protocol: -----> {sip}:
sendCallProceedingTone: -----> {false}:
rtcpEnabled: -----> {false}:
rtcpPacketInterval: -----> {5000}:
interdigitTimeOut: -----> {10}:
ipTos: -----> {0}:
systemDomainName: -----> {}:
expires-invite-value: -----> {3600}:
expires-register-value: -----> {3600}:
expires-header-method: -----> {register}:
session-timer: -----> {off}:
session-expiration: -----> {180}:
session-min-session-expiration: ---> {180}:
session-caller-request-timer: -----> {no}:
session-callee-request-timer: -----> {no}:
session-caller-specify-refresher: -> {omit}:
session-callee-specify-refresher: -> {uac}:
dtmf-mode: -----> {rfc2833}:
rtp-termid-syntax: -----> {}:
rtpDSCP:-----> (0):
signalingDSCP:-----> (0):
dtmf-payload-id: -----> {101}:
register-ready-timeout: -----> {10}:
```



```

message-retry-count: -----> {1}:
voip-features: -----> {NONE(0)}:
transport-protocol: -----> {udp}:
signalling-local-port-number: -----> {5060}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s

```

ESA for SIP PLAR

This section describes ESA SIP support on the MXK.

For VoIP SIP PLAR voice connections, the MXK provides emergency calling services during network or equipment failures that cause a loss of connection to the configured SIP server or voice gateway MALC.

For VoIP SIP PLAR connections, the ESA feature enables numbers configured within ESA dialplans to communicate with any residences or businesses specified as the destination of the dialplans in an ESA cluster of MXK devices. Incoming calls from outside the ESA group and outgoing calls to numbers outside the ESA cluster receive a fast-busy signal.

When ESA is activated, call features such as call waiting, are not supported.

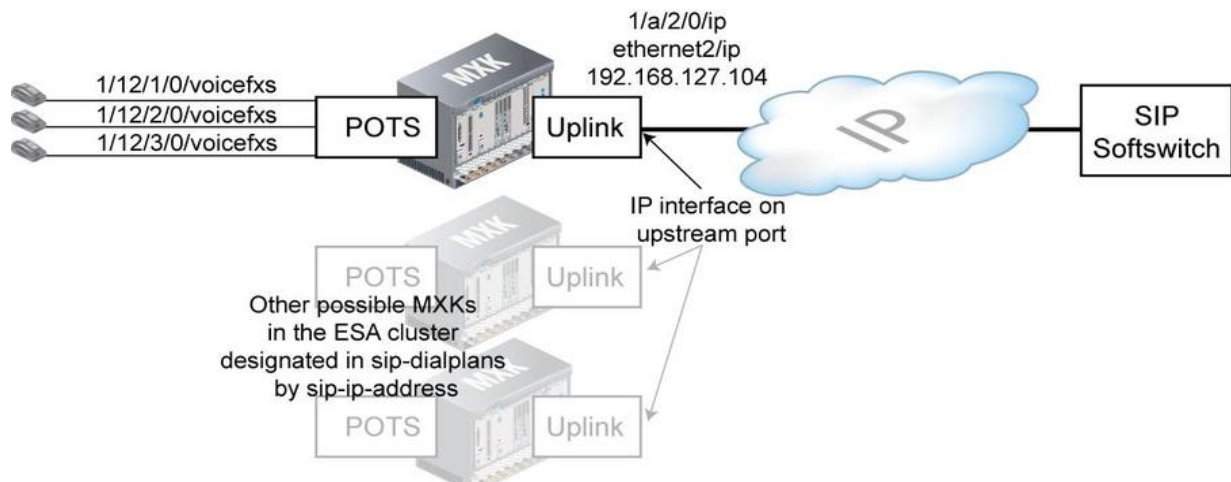


Note: After a loss of connection to the SIP PLAR server, there may be a delay up to 5 minutes before ESA notification is received and ESA features are accessible.

There may be a similar delay before resuming normal calling after the outage is restored.

Figure 64 illustrates ESA support for VoIP SIP PLAR connections.

Figure 65: ESA for VoIP SIP PLAR connections



Configuring ESA for SIP PLAR

VoIP ESA cluster requires an ESA SIP dialplan in each of the SLMS device that participate in the ESA cluster. One MXK For each ESA dialplan, enter

the IP addresses of the desired MXK in the **sip-ip-address** field and change the **dialplan-type** to **esa**. Also, if desired, change the **destination-name** to the target MXK.

When in the ESA mode, the MXK sequentially checks the configured dialplans for a matching string starting with the lowest number to the highest number dialplan. If a match is found, the call connection process is initiated immediately. If a match is not found, the next sequential dialplan is checked until all configured dialplans have been checked. Calls with unmatched strings are then terminated. It is recommended to configure lower number dialplans for more frequently called nodes and higher number dialplans for less frequently called nodes.

This example creates SIP dialplans for MXK devices. SIP dialplan 1 is used on MXK 1 with IP address 172.24.94.219; SIP dialplan 2 is used on MXK 2 with IP address 172.24.94.222. SIP dialplan 3 is used on MXK 3 with IP address 172.24.94.223. It also sets the match-string to '*x.T | x.T' to accept all numbers, all number of digits, and the dialplan type to ESA. This dialplan enables ESA calls to connect to other subscribers within the same MXK. Additional dialplans are created for each of the neighboring MXK nodes.



Note: Configuring ESA for SIP PLAR does not required to create a SIP dialplan of type normal for non-ESA calls.

- 1 Create a SIP dialplan for MXK #1. Make sure the voip-server-entry-index is 0:

```
zSH> new sip-dialplan 1
match-string: -----> {0} *x.T | x.T
sip-ip-address: -----> {0} 172.24.94.219
destination-name: -----> {}
number-of-digits: -----> {0}
prefix-strip: -----> {0}
prefix-add: -----> {}
dialplan-type: -----> {normal}esa
voip-server-entry-index: -----> {0}
override-interdigit-timeout: -> {0}
description: -----> {}:
dialplan-class: -----> {NONE(0)}:
```

- 2 Create additional SIP dialplans for so ESA calls can connect to subscribers on other SLMS devices. This dialplan allows ESA calls to connect to subscribers on MXK #2.

```
zSH> new sip-dialplan 2
match-string: -----> {0} *x.T | x.T
sip-ip-address: -----> {0} 172.24.94.222
destination-name: -----> {}
number-of-digits: -----> {0}
prefix-strip: -----> {0}
prefix-add: -----> {}
dialplan-type: -----> {normal}esa
voip-server-entry-index: -----> {0}
```

```

override-interdigit-timeout: -> {0}
description: -----> {}:
dialplan-class: -----> {NONE(0)}:

```

3 This dialplan allows ESA calls to connect to subscribers on MXK #3.

```

zSH> new sip-dialplan 3
match-string: -----> {0} *x.T | x.T
sip-ip-address: -----> {0} 172.24.94.223
destination-name: -----> {}
number-of-digits: -----> {0}
prefix-strip: -----> {0}
prefix-add: -----> {}
dialplan-type: -----> {normal}esa
voip-server-entry-index: -----> {0}
override-interdigit-timeout: -> {0}
description: -----> {}:
dialplan-class: -----> {NONE(0)}:

```

4 To configure ESA for SIP PLAR connections for 911 calls, create an ESA dialplan with a match-string of 911 and the IP address of the MXK shelf in the **sip-ip-address** field. Also, change the **prefix-strip** to 3. The prefix-strip setting deletes the dialed 911 numbers. Enter the desired phone number to be called in the **prefix-add** field. This number must be a valid voicefxs line in the same MXK shelf. Change the **dial-plan type** to **esa**.

```

zSH> new sip-dialplan 911
match-string: -----> {}911
sip-ip-address: -----> {0} 172.24.94.219
destination-name: -----> {}
number-of-digits: -----> {0}
prefix-strip: -----> {0}3
prefix-add: -----> {}7281001
dialplan-type: -----> {normal}esa
voip-server-entry-index: -----> {0}
override-interdigit-timeout: -> {0}
description: -----> {}:
dialplan-class: -----> {NONE(0)}:

```

5 Enter the **voice status** command. This command lists the voice port, destination, call state, and ESA state along with other status information

```

zSH> voice status

```

port	term	state	destination	call state	hook	ring	ESA
1-12-1-0/voicefxs		UP	VoIP:69:VoIP EndPtIdx-152	No call	ON	NoRing	
1-12-2-0/voicefxs		UP	VoIP:69:VoIP EndPtIdx-154	No call	ON	NoRing	

6 Or users can use the **sipstack esa** command.

```

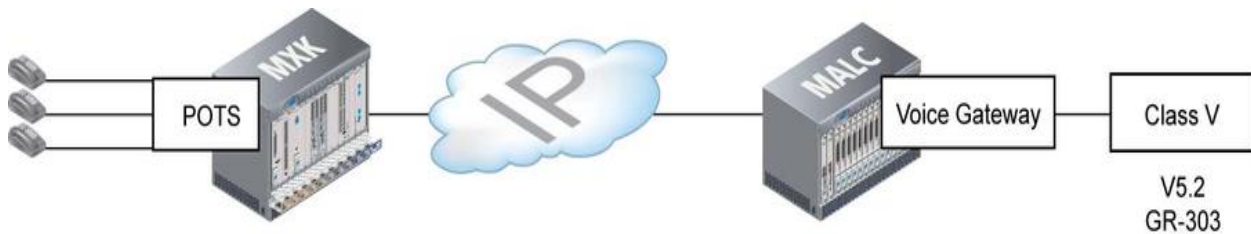
zSH> sipstack esa
sip server: 172.16.60.1:5060, Dns: 172.24.94.2 status: Not resolved # of
sub: 72 , esaMode(ip): ON

```

POTS to VoIP connection with SIP PLAR

The following figure shows for POTS-to-Voice Gateway V5.2/GR303 configuration, the feeder MXK interconnects POTS equipment to the Voice Gateway (VG) MALC, and the VG MALC connect to the class V switches (i.e. V5.2/GR 303 local exchange switches).

Figure 66: MXK common voice configuration - POTS to Class V switch



Creating POTS to VoIP connections with SIP-PLAR

The following procedure provides the VoIP configuration in the feeder MXK.

Creates a POTS to VoIP connection with SIP-PLAR signaling on the feeder MXK:

- 1 Verify/create an IP interface for voice traffic
See [Configure an IP interface for voice traffic on page 480](#).
- 2 Use the **voice add** command to add the POTS to VoIP connection. This example specifies the subscriber endpoint information to **pots 1-10-1-0/voicefxs**. The remote endpoint is refer to VG MALC, the remote endpoint information is **voip ethernet1/ip**, the directory number is **7770001**, and the ip address of VG connection is 10.6.20.2. **reg 0** means the MXK uses the SIP PLAR server that is specified in the **voip-server-entry 255/255** profile.

```

zSH> voice add pots 1-10-1-0/voicefxs voip ethernet2-100/ip dn 7770001 name
7770001 plar 10.6.20.2 reg 0 sub 7770001 enable
Created subscriber 1/3
Created subscriber-voice 1/3/1
Created subscriber-voice-pots 1
Created subscriber-voice-voip 2

```

- 3 View the voice connection.

```

zSH> voice show

```

Subscriber end-point	Remote End point	Username	SRV STA
1-10-1-0/voicefxs	ethernet1/ip	7770001	0 ENA

Total number of voice connections : 1

- 4 Use the **voice ring** command to verify a POTS voice connection by ringing the phone. The **voice status** command can be used to display runtime voice port status, verify the phone's ring status if the ringing cannot be heard, and display interface group status.

Creating POTS to VG connections with SIP-PLAR

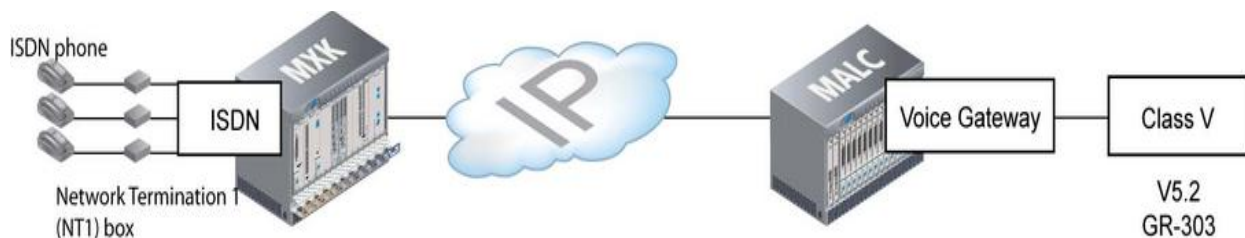
In this example, a MALC with voice gateway card receives the VoIP signal and send it to Class V switch as either an GR-303 or V5.2 voice signal.

For the VoIP configuration in the VG MALC side, refer to the *MALC Configuration Guide*.

ISDN to VoIP connection with SIP PLAR

The following figure shows for ISDN-to-Voice Gateway V5.2/GR303 configuration, the feeder MXK interconnects ISDN equipment to the Voice Gateway (VG) MALC, and the VG MALC connect to the class V switches (i.e. V5.2/GR 303 local exchange switches).

Figure 67: MXK common voice configuration - ISDN to Class V switch



Creating ISDN to VoIP connections with SIP-PLAR

The following procedure provides the VoIP configuration in the feeder MXK.

Creates a ISDN to VoIP connection with SIP-PLAR signaling on the feeder MXK:

- 1 Verify/create an IP interface for voice traffic.
See [Configure an IP interface for voice traffic on page 480](#).
- 2 Use the **voice add** command to add the ISDN to VoIP connection. This example specifies the subscriber endpoint information to **isdn 1-12-3-0/isdnu**. The remote endpoint is refer to VG MALC, the remote endpoint information is **voip ethernet5-94/ip**, the directory number and name are **0141800002**, and the ip address of VG connection is 172.25.138.2. **reg 0** means the MXK uses the SIP PLAR server that is specified in the **voip-server-entry 255/255** profile.

```
zSH> voice add isdn 1-12-3-0/isdnu voip ethernet5-94/ip dn 0141800002 name
0141800002 plar 172.25.138.2 reg 0
Created subscriber-voice 1/11/34
Created subscriber-voice-isdn 243
Created subscriber-voice-voip 244
Created subscriber-voice 1/11/35
Created subscriber-voice-isdn 245
Created subscriber-voice-voip 246
Created subscriber-voice 1/11/36
Created subscriber-voice-isdn 247
Created subscriber-voice-voip 248
```

- 3 View the details of the voice connection. Each voice add command for ISDN 2B1Q card creates three voice connections: 1. ISDN to VoIP/DN; 2. ISDN to B1; 3. ISDN to B2.

```
zSH> voice show -v
Subscriber end-point      Remote End point      Username      SRV
STA  Voice Prof Id  DN
-----
1-12-3-0/isdnu          ethernet5-94/ip          0141800002      0
ENA  1/11/34          0141800002
1-12-3-0/isdnu          ethernet5-94/ip          0141800002/b1    0
ENA  1/11/35          0141800002-1
1-12-3-0/isdnu          ethernet5-94/ip          0141800002/b2    0
ENA  1/11/36          0141800002-2
Total number of voice connections : 3
```

- 4 You can use the **voice status** command to display runtime voice port status, verify the phone’s ring status if the ringing cannot be heard, and display interface group status.

Creating POTS to VG connections with SIP-PLAR

In this example, a MALC with voice gateway card receives the VoIP signal and send it to Class V switch as either an GR-303 or V5.2 voice signal.

For the VoIP configuration in the VG MALC side, refer to the *MALC Configuration Guide*.

MGCP

- [MGCP server on page 507](#)
- [POTS to VoIP connection with MGCP on page 509](#)

MGCP server

MGCP signaling establishes call control elements or call agents to handle call control. MGCP devices execute the commands sent by the call agents.

The MXK can support redundant MGCP servers per VoIP system. In order to support multiple MGCP servers, the servers must be configured as redundant MGCP servers with redundant peer support enabled.

During the MXK system boot up, the MXK determines which redundant MGCP server use.

Configuring redundant MGCP servers

To support multiple MGCP servers, create a **voip-server-entry** *serverID/addressIndex* profile for each MGCP server. For example, 1/2 means server ID 1 and address index 2. The redundant MGCP server must use the same server ID as the primary MGCP server.

This example creates voip-server-entry profiles for two MGCP servers using server ID 1 and address indexes 1 and 2 with the keyword **mgcp** in the **protocol** field.



Note: The MGCP max call limiter is set at 500 calls. When the maximum number of allowable active calls is reach, the outgoing caller hears a congestion tone. For the incoming call, the phone does not ring.

To change the setting to MGCP:

- 1 Create the **voip-server-entry** profiles to enable MGCP:

```
zSH> new voip-server-entry 1/1
voip-server-entry 1/1
Please provide the following: [q]uit.
zhoneVoipServerAddrType: -----> {ipv4}:
zhoneVoipServerAddr: -----> {}: 172.16.160.1
zhoneVoipServerUdpPortNumber: -----> {5060}: 2727
zhoneVoipServerId: -----> {generic}: tekelec-t6000
protocol: -----> {sip}: mgcp
sendCallProceedingTone: -----> {false}:
rtcpEnabled: -----> {false}:
rtcpPacketInterval: -----> {5000}:
interdigitTimeOut: -----> {10}:
ipTos: -----> {0}:
systemDomainName: -----> {}:
expires-invite-value: -----> {3600}:
```

```

expires-register-value: -----> {3600}:
expires-header-method: -----> {register}:
session-timer: -----> {off}:
session-expiration: -----> {180}:
session-min-session-expiration: ---> {180}:
session-caller-request-timer: -----> {no}:
session-callee-request-timer: -----> {no}:
session-caller-specify-refresher: -> {omit}:
session-callee-specify-refresher: -> {uac}:
dtmf-mode: -----> {rfc2833}:
rtp-termid-syntax: -----> {}:
rtpDSCP:-----> (0)
signalingDSCP:-----> (0)
.....
Save new record? [s]ave, [c]hange or [q]uit: s
Record created.

```

2 Create a redundant MGCP server

```

zSH> new voip-server-entry 1/2
voip-server-entry 1/1
Please provide the following: [q]uit.
zhoneVoipServerAddrType: -----> {ipv4}:
zhoneVoipServerAddr: -----> {}: 172.16.160.3
zhoneVoipServerUdpPortNumber: -----> {5060}: 2727
zhoneVoipServerId: -----> {generic}: tekelec-t6000
protocol: -----> {sip}: mgcp
sendCallProceedingTone: -----> {false}:
rtcpEnabled: -----> {false}:
rtcpPacketInterval: -----> {5000}:
interdigitTimeOut: -----> {10}:
ipTos: -----> {0}:
systemDomainName: -----> {}:
expires-invite-value: -----> {3600}:
expires-register-value: -----> {3600}:
expires-header-method: -----> {register}:
session-timer: -----> {off}:
session-expiration: -----> {180}:
session-min-session-expiration: ---> {180}:
session-caller-request-timer: -----> {no}:
session-callee-request-timer: -----> {no}:
session-caller-specify-refresher: -> {omit}:
session-callee-specify-refresher: -> {uac}:
dtmf-mode: -----> {rfc2833}:
rtp-termid-syntax: -----> {}:
rtpDSCP:-----> (0)
signalingDSCP:-----> (0)
.....
Save new record? [s]ave, [c]hange or [q]uit: s
Record created.

```



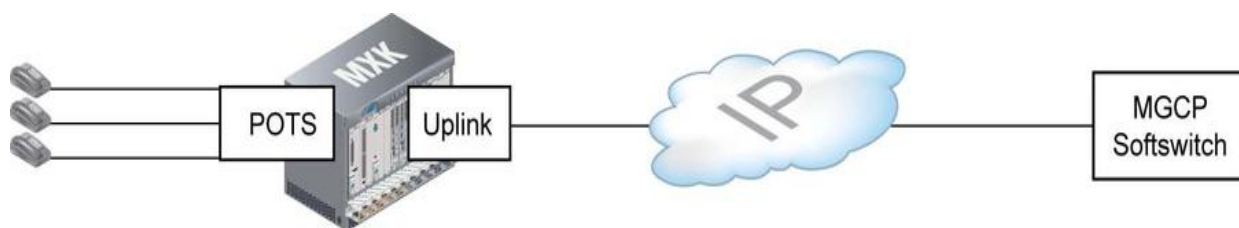

Note: The system will automatically reboot if the voice protocol is changed. After the reboot, verify that the **voip-server-entry** profile is configured for MGCP.

POTS to VoIP connection with MGCP

After configured IP interface, VoIP system, and VoIP server settings properly, user can create POTS to MGCP softswitch connections.

The following figure shows for POTS to MGCP softswitch configuration, the MXK interconnects POTS terminal equipment directly to MGCP softswitch.

Figure 68: MXK common voice configuration - POTS to MGCP Softswitch



Creating POTS to VoIP connections with MGCP

This example creates a POTS to MGCP softswitch connection:

- 1 Verify/create an IP interface for voice traffic
See [Configure an IP interface for voice traffic on page 480](#).
- 2 Verify/create the MGCP server.
See [Configuring redundant MGCP servers on page 507](#).
- 3 Use the **voice add** command to add the POTS to VoIP connection. This examples creates a connection with a directory number 201202999 and the name **aaln/1**. The VoIP remote-endpoint *user name* is case sensitive and must match the voice switch requirements.

```

zSH> voice add pots 1-10-1-0/voicefxs voip ethernet2-100/ip dn 201202999
name aaln/1 enable
Created subscriber 1/5
Created subscriber-voice 1/5/1
Created subscriber-voice-pots 1
Created subscriber-voice-voip 2
  
```

This example didn't specify the **reg** option, it means the MXK uses the primary VoIP server that is specified in the **voip-server-entry 1/x** (any address index) profile.

- 4 View the voice connection.

```
zSH> voice show
```

Subscriber end-point	Remote End point	Username	SRV STA
1-10-1-0/voicefxs	ethernet2-100/ip	aaln/1	1 ENA
Total number of voice connections : 1			

- 5 The **voice ring** command can be used to verify a POTS voice connection by ringing the phone. The **voice status** command can be used to display runtime voice port status, verify the phone’s ring status if the ringing cannot be heard, and display interface group status.

H.248

- [H.248 configuration on page 511](#)
- [POTS to VoIP connection with H.248 on page 512](#)
- [ISDN to VoIP connection with H.248 on page 513](#)
- [ESA for H.248 on page 514](#)

H.248 configuration

The H.248 protocol is used between elements of a physically decomposed multimedia gateway. The distributed multimedia gateway sub-components create a general framework used for gateways, multipoint control units and interactive voice response units (IVRs).

Configuring H.248

This example creates **voip-server-entry serverID/address Index** profiles for a H.248 VoIP server using server ID 1 and address Index 1 with keyword **megaco** in the **protocol** field.

This example keeps default value 10 seconds in the **register-ready-timeout** field. This field is used for Megaco service change messages. The value is in the range of 0 ... 4294967295, the max of 32 bit integer.

Create the **voip-server-entry** profiles to enable H.248

```
zSH> new voip-server-entry 1/1
voip-server-entry 1/1
Please provide the following: [q]uit.
zhoneVoipServerAddrType: -----> {ipv4}:
zhoneVoipServerAddr: -----> {}: 172.16.160.1
zhoneVoipServerUdpPortNumber: -----> {5060}: 2944
zhoneVoipServerId: -----> {generic}: nortel-cs2000
protocol: -----> {sip}: megaco
sendCallProceedingTone: -----> {false}:
rtcpEnabled: -----> {false}:
rtcpPacketInterval: -----> {5000}:
interdigitTimeOut: -----> {10}:
ipTos: -----> {0}:
systemDomainName: -----> {}:
expires-invite-value: -----> {3600}:
expires-register-value: -----> {3600}:
expires-header-method: -----> {register}:
session-timer: -----> {off}:
session-expiration: -----> {180}:
session-min-session-expiration: ---> {180}:
session-caller-request-timer: -----> {no}:
session-callee-request-timer: -----> {no}:
session-caller-specify-refresher: -> {omit}:
session-callee-specify-refresher: -> {uac}:
dtmf-mode: -----> {rfc2833}:
```

```

rtp-termid-syntax: -----> {}:
rtpDSCP: -----> {0}:
signalingDSCP: -----> {0}:
dtmf-payload-id: -----> {101}:
register-ready-timeout: -----> {10}:
message-retry-count: -----> {1}:
voip-features: -----> {NONE(0)}:
transport-protocol: -----> {udp}:
signalling-local-port-number: -----> {5060}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s

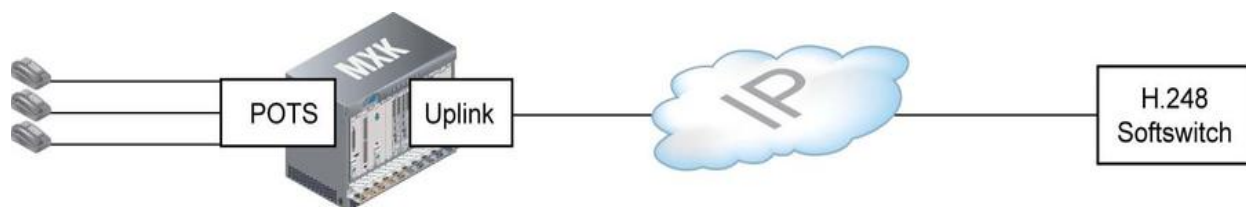
```

POTS to VoIP connection with H.248

After configured IP interface, VoIP system, and VoIP server settings properly, user can create POTS to H.248 softswitch connections.

The following figure shows for POTS to H.248 softswitch configuration, the MXK interconnects POTS terminal equipment directly to H.248 softswitch.

Figure 69: MXK common voice configuration - POTS to H.248 Softswitch



Creating POTS to VoIP connections

This example creates a POTS to VOIP subscriber:

- 1 Verify/create an IP interface for voice traffic
See [Configure an IP interface for voice traffic on page 480](#).
- 2 Verify/create the H.248 server.
See [Configuring H.248 on page 511](#).
- 3 Use the **voice add** command to add the POTS to VoIP connection. This examples creates a connection with a directory number 201202999 and the name **tp/0000**. The VoIP remote-endpoint *user name* is case sensitive and must match the voice switch requirements.

```

zSH> voice add pots 1-10-1-0/voicefxs voip ethernet2-100/ip dn 201202999
name tp/0000 enable
Created subscriber 1/5
Created subscriber-voice 1/5/1
Created subscriber-voice-pots 1
Created subscriber-voice-voip 2

```

This example didn't specify the **reg** option, it means the MXK uses the primary VoIP server that is specified in the **voip-server-entry 1/x** (any address index) profile.

4 View the voice connection.

```
zSH> voice show
Subscriber end-point      Remote End point      Username      SRV STA
-----
1-10-1-0/voicefxs        ethernet2-100/ip      tp/0000      1 ENA
Total number of voice connections : 1
```

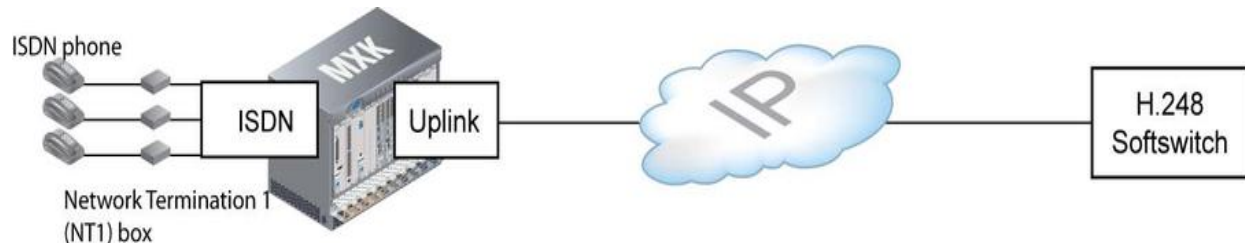
- 5** The **voice ring** command can be used to verify a POTS voice connection by ringing the phone. The **voice status** command can be used to display runtime voice port status, verify the phone's ring status if the ringing cannot be heard, and display interface group status.

ISDN to VoIP connection with H.248

After configured IP interface, VoIP system, and VoIP server settings properly, user can create ISDN to H.248 softswitch connections.

The following figure shows for ISDN to H.248 softswitch configuration, the MXK interconnects ISDN terminal equipment directly to H.248 softswitch.

Figure 70: MXK common voice configuration - ISDN to H.248 Softswitch



Creating ISDN to H.248 connections

This example creates a ISDN to H.248 subscriber:

- 1 Verify/create an IP interface for voice traffic.
See [Configure an IP interface for voice traffic on page 480](#).
- 2 Verify/create the H.248 server.
See [Configuring H.248 on page 511](#).
- 3 Create the IUA server. This step is required for H248 configuration to resolve the IUA for ISDN.

```
zSH> new iua-server-entry 1/1
iua-server-entry 1/1
Please provide the following: [q]uit.
iua-server-addr: -----> {}: 172.16.171.1
```

```
iua-server-port-number: -> {9900}:
.....
```

```
Save new record? [s]ave, [c]hange or [q]uit:s
```

- 4 Use the **voice add** command to add the ISDN to H.248 connection. This example creates a connection with a directory number **9029824960** and the name **ba/0**. The VoIP remote-endpoint *user name* is case sensitive and must match the voice switch requirements.

```
zSH> voice add isdn 1-14-3-0/isdn voip ethernet2-959/ip dn 9029824960 name ba/0
Created subscriber-voice 1/5/16
Created subscriber-voice-isdn 31
Created subscriber-voice-voip 32
Created subscriber-voice 1/5/17
Created subscriber-voice-isdn 33
Created subscriber-voice-voip 34
Created subscriber-voice 1/5/18
Created subscriber-voice-isdn 35
Created subscriber-voice-voip 36
```

This example didn't specify the **reg** option, it means the MXK uses the primary VoIP server (reg 1) that is specified in the **voip-server-entry 1/x** (any address index) profile.

- 5 View the voice connection. Each voice add command for ISDN 2B1Q card creates three voice connections: 1. ISDN to VoIP/DN; 2. ISDN to B1; 3. ISDN to B2.

```
zSH> voice show
```

Subscriber end-point STA	Remote End point	Username	SRV
1-14-3-0/isdn ENA	ethernet2-959/ip	ba/0	1
1-14-3-0/isdn ENA	ethernet2-959/ip	ba/0/b1	1
1-14-3-0/isdn ENA	ethernet2-959/ip	ba/0/b2	1

```
Total number of voice connections : 3
```

- 6 You can use the **voice status** command to display runtime voice port status, verify the phone's ring status if the ringing cannot be heard, and display interface group status.

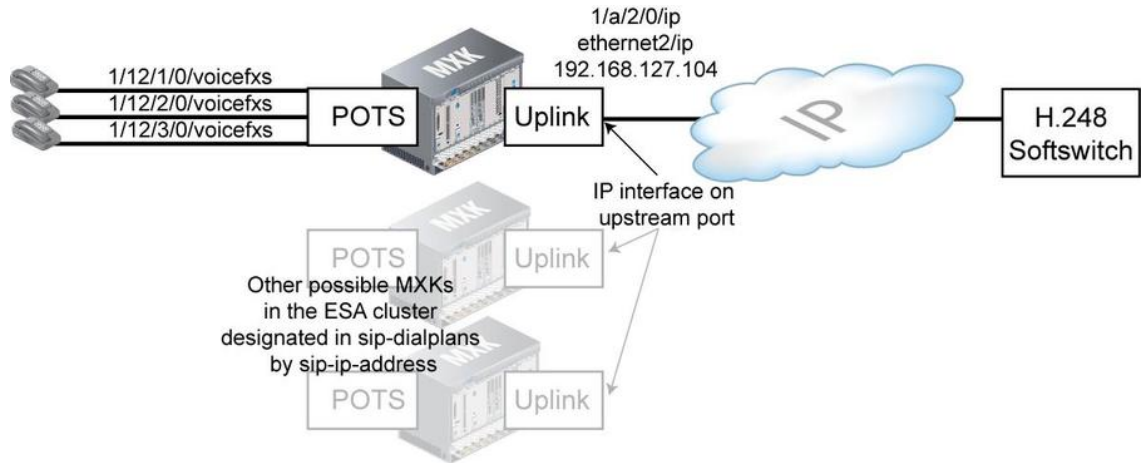
ESA for H.248

Just as with SIP ESA, if the MXK loses H.248 communication with the softswitch, the MXK will continue to process calls locally between subscribers in the same MXK chassis to another reachable MXK in the ESA cluster. POTS subscribers on the same MXK can make calls (voice, fax, modem) between each other as well as calls to other reachable MXKs in the

ESA cluster, based on the predefined dial plans for each MXK in the ESA cluster.

Since communication to the softswitch server is lost, there is no communication outside the ESA cluster.

Figure 71: ESA for H.248 softswitch



When the H.248 communication to the softswitch is lost, the MXK waits for the time configured in the no-response-timer in the voice-system profile, then switches to ESA mode. (see [Configuring ESA timers, page 520](#)). The same timer is used for switching back from ESA mode when the MXK detects the connection to the H.248 switch has returned. All SIP ESA functionality is supported. To go into SIP, ESA dialplans identify the IP address of the participating MXKs in the ESA cluster.

To configure ESA for H.248 create a SIP dialplan for each MXK in the ESA cluster using the MXK's IP address with the digitmap “***x.T** | **x.T**” as shown in the procedure. Each MXK in the cluster will be tried when in ESA mode.

Configuring ESA for H.248

While it only takes the three steps: creating the two voip-server-entries and the sip-dialplan(s) to configure the MXK for POTS ESA for H.248, this procedure also shows verification steps, so you can analyze existing configurations.

Note that if you already have a primary **voip-server-entry** (For example 1/1 or 2/1, or 3/1 etc.) with protocol as **megaco**, then you only need to create additional **voip-server-entry** with **sip** protocol for the ESA fallback and the sip-dialplans(s)

To differentiate the two voip-server-entries the key is to compare the voip-server-entry address. The **voip-server-entry** with address index **1**, for example 1/1 or 2/1 with protocol **megaco** will be always be considered as the primary **voip-server-entry** and the voip-server address with the index greater than **1** (with **sip** protocol) will be considered as backup entry. For example for

primary **voip-server-entry 1/1**, 1/<any> with protocol set to SIP will be considered the backup entry.

1 Verify or create interface for uplink.

```
zSH> interface show
```

```
2 interfaces
```

Interface	Status	Rd/Address	Media/Dest Address	IfName
1/a/1/0/ip	UP	1 172.24.200.50/24	00:01:47:2b:c2:c0	ethernet1
1/a/2/0/ip	UP	1 192.168.127.104/24	00:01:47:2b:c2:c7	ethernet2

Notice the IP address and the interface name (IfName) on the upstream interface.

2 Create the **voip-server-entry** to H.248 softswitch.

```
zSH> new voip-server-entry 1/1
```

```
voip-server-entry 1/1
```

```
Please provide the following: [q]uit.
```

```
zhoneVoipServerAddrType: -----> {ipv4}:
zhoneVoipServerAddr: -----> {}: 172.60.0.65
zhoneVoipServerUdpPortNumber: -----> {5060}: 2944
zhoneVoipServerId: -----> {generic}:
nortel-cs2000
protocol: -----> {sip}: megaco
sendCallProceedingTone: -----> {false}:
rtcpEnabled: -----> {false}:
rtcpPacketInterval: -----> {5000}:
interdigitTimeOut: -----> {10}:
ipTos: -----> {0}:
systemDomainName: -----> {}:
expires-invite-value: -----> {3600}:
expires-register-value: -----> {3600}:
expires-header-method: -----> {register}:
session-timer: -----> {off}:
session-expiration: -----> {180}:
session-min-session-expiration: ---> {180}:
session-caller-request-timer: -----> {no}:
session-callee-request-timer: -----> {no}:
session-caller-specify-refresher: -> {omit}:
session-callee-specify-refresher: -> {uac}:
dtmf-mode: -----> {rfc2833}:
rtp-termid-syntax: -----> {}:
rtpDSCP: -----> {0}:
signalingDSCP: -----> {0}:
dtmf-payload-id: -----> {101}:
register-ready-timeout: -----> {10}:
message-retry-count: -----> {1}:
voip-features: -----> {NONE(0)}:
transport-protocol: -----> {udp}:
```



```

signalling-local-port-number: -----> {5060}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```

The first index is for the H.248 connection which points to the H.248 server (The `zhoneVoipServerAddr` parameter is 172.60.0.65 in the example). **2944** is the UDP port for H.248. The **protocol** must be **megaco**.

3 Create **voip-server-entry** for SIP which is used for the ESA clusters

```
zSH> new voip-server-entry 1/2
```

```

voip-server-entry 1/2
Please provide the following: [q]uit.
zhoneVoipServerAddrType: -----> {ipv4}:
zhoneVoipServerAddr: -----> {}: 0.0.0.0This
setting for the backup entry should always be set to
"0.0.0.0"
zhoneVoipServerUdpPortNumber: -----> {5060}: This
setting for the backup entry should always be set to
"5060" the UDP port for SIP
zhoneVoipServerId: -----> {generic}:
protocol: -----> {sip}: This
setting for the backup entry should always be set to
"sip"

sendCallProceedingTone: -----> {false}:
rtcpEnabled: -----> {false}:
rtcpPacketInterval: -----> {5000}:
interdigitTimeout: -----> {10}:
ipTos: -----> {0}:
systemDomainName: -----> {}:
expires-invite-value: -----> {3600}:
expires-register-value: -----> {3600}:
expires-header-method: -----> {register}:
session-timer: -----> {off}:
session-expiration: -----> {180}:
session-min-session-expiration: ---> {180}:
session-caller-request-timer: -----> {no}:
session-callee-request-timer: -----> {no}:
session-caller-specify-refresher: -> {omit}:
session-callee-specify-refresher: -> {uac}:
dtmf-mode: -----> {rfc2833}:
rtp-termid-syntax: -----> {}:
rtpDSCP: -----> {0}:
signalingDSCP: -----> {0}:
dtmf-payload-id: -----> {101}:
register-ready-timeout: -----> {10}:
message-retry-count: -----> {1}:
voip-features: -----> {NONE(0)}:
transport-protocol: -----> {udp}:
signalling-local-port-number: -----> {5060}:

```

```

.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```

Since SIP is the default with **protocol = sip** and the UDP port = **5060**, all you need do is create the second subindex (1/2) for this backup entry; the primary H.248 **voip-server-profile** is index 1/1.

4 Add the ESA sip-dialplan(s)

This example creates a SIP dialplan for so ESA calls can connect to subscribers on MXK 1 with 172.24.94.219:

```

zSH> new sip-dialplan 1
sip-dialplan 1
Please provide the following: [q]uit.
match-string: -----> {}: 55511xx
sip-ip-address: -----> {0.0.0.0}:172.24.94.219
destination-name: -----> {}:
number-of-digits: -----> {0}:
prefix-strip: -----> {0}:
prefix-add: -----> {}:
dialplan-type: -----> {normal}: esa
voip-server-entry-index: -----> {0}:
override-interdigit-timeout: -> {0}:
dialplan-class: -----> {NONE(0)}:
description: -----> {}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```

Create a SIP dialplan for so ESA calls can connect to subscribers on MXK 2:

```

zSH> new sip-dialplan 2
sip-dialplan 2
Please provide the following: [q]uit.
match-string: -----> {}: 55512xx
sip-ip-address: -----> {0.0.0.0}:172.24.94.222
destination-name: -----> {}:
number-of-digits: -----> {0}:
prefix-strip: -----> {0}:
prefix-add: -----> {}:
dialplan-type: -----> {normal}: esa
voip-server-entry-index: -----> {0}:
override-interdigit-timeout: -> {0}:
dialplan-class: -----> {NONE(0)}:
description: -----> {}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```

Create a SIP dialplan 911 on the MXK 1. It replaces the dialed 911 number with the phone number 7281001 and changes the dialplan type to ESA:

```
zSH> new sip-dialplan 911
sip-dialplan 3
Please provide the following: [q]uit.
match-string: -----> {}: 911
sip-ip-address: -----> {0.0.0.0}:172.24.94.219
destination-name: -----> {}:
number-of-digits: -----> {0}:
prefix-strip: -----> {0}: 3
prefix-add: -----> {}: 7281001
dialplan-type: -----> {normal}: esa
voip-server-entry-index: -----> {0}:
override-interdigit-timeout: -> {0}:
dialplan-class: -----> {NONE(0)}:
description: -----> {}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Creating the sip-dial plan as shown above, does not make ESA mode on. Creating the sip-dial plan which creates the configuration to route the calls when the MXK is in ESA mode.

5 Verify or create POTS interfaces

```
zSH> voice add pots 1-12-1-0/voicefxs voip ethernet2/
ip dn 201749 name tp/0000 enable
Created subscriber-voice 12/5/1
Created subscriber-voice-pots 1
Created subscriber-voice-voip 2
```

```
zSH> voice add pots 1-12-2-0/voicefxs voip ethernet2/
ip dn 576006 name tp/0000 enable
Created subscriber-voice 12/5/2
Created subscriber-voice-pots 3
Created subscriber-voice-voip 4
```

```
zSH> voice add pots 1-12-3-0/voicefxs voip ethernet2/
ip dn 208119 name tp/0000 enable
Created subscriber-voice 12/5/3
Created subscriber-voice-pots 5
Created subscriber-voice-voip 6
```

Notice the **interface/type** for the uplink.

6 View voice connections

The **voice show -v** command shows the voice connections.

```
zSH> voice show -v
Subscriber end-pointRemote End pointUsernameSRV STA
Voice Prof IdDN
```

```

-----
-----
-----
1-12-1-0/voicefxsethernet2/iptp/00001 ENA1/5/1201749
1-12-2-0/voicefxsethernet2/iptp/00001 ENA1/5/2576006
1-12-3-0/voicefxsethernet2/iptp/00001 ENA1/5/3208119
Total number of voice connections : 3

```

- a** After configuring ESA for H.248, ESA mode can be verified by using the **esa voip show mode** command.

```

zSH> esa voip show mode
Esa is OFF

```

- b** H.248 server information can be verified by using the **megacostack server** command.

```

zSH> megacostack server
VOIP SERVER INFO:
~~~~~
Valid -----> TRUE
ClockHdl -----> 0x0
Server Addr -----> 172.60.0.65
Server Port -----> 2944
assocId -----> 1
AssocState -----> 3
Server Contact -----> Responded
Server Response Miss-> 0
Server is AUEP Mon --> ON
Message received ----> TRUE
ITO active -----> TRUE
ITO value -----> 1200
ESA Feature -----> Enabled
ESA Mode -----> OFF
ESA Auto switch ----> ON
ESA Auto switchback -> ON
Keep Alive Timer Interval: 60 sec
KeepAliveRunning :-----> TRUE
RSIP Retry Timer :-----> Off
KeepAlive Timer :-----> On

```

- c** SIP server information can be verified by using the **sipstack esa** command.

```

zSH> sipstack esa
sip server: 0.0.0.0:5060, Dns: 172.24.94.2 status:
Not resolved # of sub: 72 , esaMode(ip): OFF

```

Configuring ESA timers

Update the no-response-timer (in seconds)

```

zSH> update voice-system 0

```

```

voice-system 0
Please provide the following: [q]uit.
hookflash-min-timer: -----> {100}:
hookflash-max-timer: -----> {1550}:
partial-dial-timeout: -----> {16}:
critical-dial-timeout: -----> {4}:
busy-tone-timeout: -----> {30}:
dial-tone-timeout: -----> {16}:
msg-wait-tone-timeout: -----> {16}:
offhook-warn-tone-timeout: -> {0}:
ringing-timeout: -----> {180}:
ringback-timeout: -----> {180}:
reorder-tone-timeout: -----> {30}:
stutter-tone-timeout: -----> {16}:
server-max-timer: -----> {20}:
config-max1: -----> {5}:
config-max2: -----> {7}:
max1-enable: -----> {true}:
max2-enable: -----> {true}:
max-waiting-delay: -----> {600}:
disconnection-wait-timer: --> {15}:
disconnection-min-timer: ---> {15}:
disconnection-max-timer: ---> {600}:
max-retransmit-timer: -----> {4}:
init-retransmit-timer: -----> {200}:
keep-alive-timer: -----> {60}:
no-response-timer: -----> {30}: 20
call-wait-max-repeat: -----> {2}:
call-wait-delay: -----> {10}:
pulse-inter-digit-timer: ---> {240}:
min-make-pulse-width: -----> {15}:
max-make-pulse-width: -----> {55}:
min-break-pulse-width: -----> {35}:
max-break-pulse-width: -----> {75}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Subscriber voice features configuration

This section describes the configurable subscriber voice features for VoIP-enabled services.

You can modify the **features** parameter in the **subscriber-voice** profile to add more VoIP features for the subscriber, such as call transfer or local conferencing.

After changing the feature settings, use the **voice bounce** command to disable and then enable the voice-admin-status for this change to take effect.

If you want to set features while creating the POTs and VoIP connection, use the **voice add** command plus the feature options.

- [Default subscriber voice features, page 522](#)
- [Call transfer, page 524](#)
- [SIP local call conferencing, page 525](#)
- [SIP local intercom, page 527](#)
- [Line Side Answer Supervision and reverse battery signal support for payphones, page 530](#)
- [DTMF mode support per port basis, page 533](#)
- [Data exchange only, page 535](#)
- [Voice exchange only, page 536](#)
- [Plar, page 537](#)
- [Hotline and Warmline, page 538](#)
- [Cut-off on Disconnect, page 539](#)
- [Always off hook, page 540](#)
- [Centrex, page 541](#)

Default subscriber voice features

The default subscriber features are hookflash, on-hook signaling, and call waiting. These features are implemented primarily for SIP. Most MGCP and Megaco softswitches provide this type of functionality:

- Hookflash

Hookflash is either a button on the phone to simulate the quick offhook/onhook/offhook cycle or the actual cycle itself. Hookflash can be used as the trigger event for switching to call waiting or three way call conferencing.
- On-hook signaling

On-hook signaling indicates the phone can accept any features or signals that only enabled while the phone is on-hook.

- Call wait feature

When an incoming call is received the receiver of the call is notified by a tone of an incoming call; the hookflash trigger switches the subscriber between the ongoing call and the incoming call. The original call is placed on hold.

Viewing the default subscriber voice features

To view the hookflash feature:

- 1 Show the voice prof ID for the voice subscriber.

```
zSH> voice show -v
Subscriber end-point Remote End point Username SRV STA Voice Prof Id DN
-----
1-10-2-0/voicefxs ethernet2/ip tp/000 01 ENA 1/6/1 201749
Total number of voice connections : 1
```

- 2 Show the default features parameter in the subscriber-voice profile

```
zSH> update subscriber-voice 1/6/1
subscriber-voice 1/6/1
Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}: ** read-only **
voice-endpoint1-addr-index: ---> {2}: ** read-only **
voice-endpoint2-addr-index: ---> {1}: ** read-only **
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}: ** read-only **
features: -----> {hookflash+onhooksignaling+callwait}:
.....
Save changes? [s]ave, [c]hange or [q]uit: q
```

Changing the hookflash timer values

The hookflash timer values can be configured to a specified range between minimum and maximum values. If hookflash is enabled on a VoIP subscriber, a hookflash is considered only if the onhook time is between the minimum and maximum timer values. Any time less than the minimum time setting is ignored and any time more than the maximum time setting is considered to be onhook.

[Table 43](#) describes the hookflash configurable timer settings in the **voice-system 0**.

Table 43: hookflash timer parameter values

Parameter	Description
hookflash-min-timer	Specifies the minimum hookflash timer value in milliseconds. Values: 0 to 2147483647 Default: 100 milliseconds
hookflash-max-timer	Specifies the maximum hookflash timer value in milliseconds. Values: 0 to 2147483647 Default: 1550 milliseconds

To change the hookflash timer values:

```
zSH> update voice-system 0
Please provide the following: [q]uit.
hookflash-min-timer: -> {100}: 500
hookflash-max-timer: -> {1550}: 2000
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Call transfer

When the call transfer feature is added to hookflash, the MXK supports transferring calls. The hookflash trigger during an ongoing call gives the subscriber a secondary dialtone and will accept dialing. The original call is on hold until another hookflash.

Adding call transfer

To add the call transfer feature:

- 1 Show the voice prof ID for the voice subscriber.

```
zSH> voice show -v
Subscriber end-pointRemote End pointUsernameSRV STA
Voice Prof IdDN
-----
-----
-----
1-10-2-0/voicefxs ethernet2/ipZ9997/04011 ENA1/4/1
201749
Total number of voice connections : 1
```

- 2 Update the features parameter in the subscriber-voice profile

```
zSH> update subscriber-voice 1/4/1
subscriber-voice 1/4/1
```



```

Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}: ** read-only **
voice-endpoint1-addr-index: ---> {2}: ** read-only **
voice-endpoint2-addr-index: ---> {1}: ** read-only **
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}: ** read-only **
features: -----> {hookflash+onhooksignaling+callwait}:
hookflash+onhooksignaling+callwait+calltransfer
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

3 Bounce the voice port for the feature to take effect.

```
zSH>voice bounce 1-10-2-0/voicefxs
```

Bouncing the port disables then enables the connection, so that the added feature will take effect.

SIP local call conferencing

The MXK local call conferencing feature is supported only with SIP.

MGCP and H.248 have the conferencing feature on their switch side.

The MXK call conferencing feature enables three-way conference calls during which three parties can use one calling session to communicate. The voice cards support call conferencing. These cards work with any VOIP-enabled uplink card installed in the MXK.

The MXK call conferencing feature deploys an efficient end-mixing conference call technology, avoiding the overhead of the centralized conference server.

Three-way call conferencing follows the Telcordia (Bellcore) three-way calling standard called Telcordia - TR - TSY - 000577, Three-Way Calling.

Configuring call conferencing on the MXK

The call conference feature is enabled through the **features** parameter in the **subscriber-voice** profile for callers using the specified port on a MXK voice card. By default, this feature is disabled.

To enable conferencing, use the **voice show -v** command to identify voice profile ID for the desired voice subscriber. Then, update the **subscriber-voice** profile for the desired subscriber with support for hookflash and conference. Additional features such as onhooksignaling and call waiting can also be added.

The following example configures call conferencing along with onhooksignaling and call waiting for the voice subscriber 1/3/1.

1) show the voice profile ID for the voice subscriber.

```

zSH> voice show -v
Subscriber end-point Remote End point Username SRV STA Voice Prof IdDN
-----
1-10-2-0/voicefxs ethernet2/ip Z9997/0401 1 ENA 1/3/1 201749
Total number of voice connections : 1

```

2) Configure call conferencing along with onhooksignalling and call waiting for the voice subscriber 1/3/1.

```

zSH> update subscriber-voice 1/3/1
Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}: ** read-only **
voice-endpoint1-addr-index: ---> {2}: ** read-only **
voice-endpoint2-addr-index: ---> {1}: ** read-only **
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}: ** read-only **
features: -----> {hookflash+onhooksignaling+callwait}:
hookflash+onhooksignaling+callwait+conference
.....
Save changes? [s]ave, [c]hange or [q]uit: s

```

3) Bounce the voice port for the feature to take effect.

```
zSH>voice bounce 1-10-2-0/voicefxs
```

Bouncing the port disables then enables the connection, so that the added feature will take effect.

Connecting three-way conference calls

The process of connecting a three-way conference call involves the following steps:

1. Caller dials the phone number of the first conference participate.
This establishes a two-way speech path between the caller and the first participate.
2. After establishing the call, the caller presses the **Flash** button or provides hookflash.
This place the first participate on hold and sends a hookflash signal to the MXK for a second dial tone.
3. Caller dials the phone number of the second conference participate.
This establishes a two-way speech path between the caller and the second participate.
4. After establishing the second call, the caller presses the **Flash** button or provides hookflash.
This establishes the three-way conference call.



Note: If the call conference features is not enabled on the MXK and a caller issues a hookflash signal while on an established call, the MXK places the current call on hold and provides a dial-tone for a second call. Subsequent hookflash signals, toggle between the two established calls.

If a hookflash signal is issued during a three-way conference call, the last conference participant is dropped and the call becomes a two-way call.

To disconnect from a three-way conference call:

- The originating caller hangs up, all members of the conference call are disconnected.
- A caller other than the originating caller hangs up, a two-way call between the originating caller and the other caller remains in progress.

Current call conferencing limitations

The following are current limitations to the call conferencing feature:

- Only SIP is supported for local call conferencing.
- The following limitation only applicable for ADSL+POTS 48 port combo card, not for POTS 72 card.

For resource utilization, three-party call conferencing divides the available 48 port resources into 8 groups of 6 sequential port resources based on physical port number (1-6, 7-12, ... ,43-48). Within a port resource group, any idle channel resource may be used for a call, including conference sessions. For a two-way call, one port resource is used. For a three-way conference call, two port resources are used.

If an idle channel resource is unavailable because of an on-going conference call within a port resource group, any new two-way call attempts receive a fast-busy tone and any three-way conference call attempts will not succeed. Three-way conference call attempts are restricted to toggling between the established two-way calls.

SIP local intercom

Intercom feature is used for subscribers who have parallel phones on the same subscriber loop. It can be used to call and converse with other parties on the same subscriber loop.

The MXK local intercom feature is supported with SIP.

This feature is local to SLMS without involving the soft switch.

Configuring SIP local intercom feature on the MXK

The SIP local intercom feature is enabled on a per voip-server basis by configuring the following fields of the **sip-dialplan** profile. By default, this feature is disabled.

- **match-string**
Specify the intercom feature activation code.
- **dialplan-type**
This field must be set to **intercom**.
- **voip-server-entry-index**
Specify the VoIP server ID for which the dialplan is used.

The following example enables SIP intercom feature for subscribers that using VoIP server 1, and the intercom feature activation code is *99.

```
zSH> new sip-dialplan 1
sip-dialplan 1
Please provide the following: [q]uit.
match-string: -----> {}: *99
sip-ip-address: -----> {0.0.0.0}:
destination-name: -----> {}:
number-of-digits: -----> {0}:
prefix-strip: -----> {0}:
prefix-add: -----> {}:
dialplan-type: -----> {normal}: intercom
voip-server-entry-index: -----> {0}: 1
override-interdigit-timeout: -> {0}:
dialplan-class: -----> {NONE(0)}:
description: -----> {}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
```

Activating and Deactivating intercom calls

After configuring intercom feature on MXK, you can follow the steps below to activate a intercom calls among the phones on the same subscriber loop:

1. Caller picks up the phone
Get the dialtone.
2. Caller dials the Intercom feature activation code.
Get the confirmation tone.
3. The originating caller hangs up.
All phones on the same line will start to ring, include the phone originating the call. The Intercom feature is in progress.
4. The first participate picks up the phone.
All the phones on the same line stop the ringing.

5. Any subscriber that on the same line picks up the phone.

The intercom call is connected.

Note that during the intercom conversation, more parties on the same subscriber loop can join by picking up the phones.

When the last phone on the line hangs up, all phones on the line are out of the intercom mode. The intercom feature is deactivated.

Interaction with other features

The following are how the intercom feature interacts with other features:

- All incoming calls will be rejected as long as the phone is in the intercom feature mode.
- VoiceMail Message Waiting Indicator (VMWI) alert will not be processed if the phone is in the intercom mode.
- Intercom feature can be only activated by dialing the Intercom feature activation code after the initial offhook. Once the initially dialed digits are processed and determined not to be Intercom feature activation code, the feature cannot be activated for the duration of the call.
- Intercom feature works in ESA mode and non-ESA mode.
- A subscriber in Intercom feature mode contributes to the total number of active calls in the system. And therefore should be considered for maximum call threshold count of the system.
- Offhook (i.e. pickup the phone) and Onhook (i.e. hang up the phone) are the only valid signals when in Intercom feature mode.
- This feature will have the ringing timeout after ringing. After ringing for 2 minutes and no one picks up, the intercom call will be disconnected.
- Inter digit timeout will be applied and feature will be deactivated if the user stays off hook after feature code activation.

The inter digit timer and the timer to wait for the user to go onhook after the user has dialed the intercom activation feature code is based on the following rules (in the order of preference):

1. Use the parameter **override-interdigit-timeout** in the **sip-dialplan** profile if it is non-zero.
 2. Use the parameter **critical-dial-timeout** in the **voice-system** profile if it is non-zero.
 3. If both of the above parameters are zero, use the hard coded timer of 4 seconds.
- Redundancy for intercom feature is not supported.

If the uplink switches over while intercom feature is in progress (i.e. when the phone is ringing due to feature activation), the ringing will stop after switchover and the phones will go back to normal mode (i.e. out of the intercom mode).

Line Side Answer Supervision and reverse battery signal support for payphones

Line Side Answer Supervision (LSAS) is a feature available on all MXK POTS-based line cards. When LSAS is enabled, an originating station on the MXK line card receives an electrical signal indicating that the terminating (called) party has answered. On the MXK, the LSAS can be either a polarity reversal of voltage (i.e. battery reversal) that the line card applies between the tip and ring conductors of the POTS line or a 12kHz/16kHz (provisionable) tone applied to the line. The most common application of LSAS is for pay phones applications to determine if and when the called party has answered the phone for billing purposes.

The MXK is capable of two kinds of indications on the local POTS subscriber when the far end answers:

- Reverse-battery

The reverse-battery feature is supported for SIP, SIP-PLAR, MGCP and H.248 softswitch applications.

For SIP, LSAS is provided when “200 OK” is received on the far end answer. The LSAS tone can be configured in the subscriber side.

For SIP-PLAR, the v5 switch configures the reverse-battery feature automatically, no configuration required at the subscriber side.

For MGCP, and H.248, the softswitch configures the reverse battery feature automatically, no configuration required at the subscriber side.

- Tone

In this case the MXK plays a far end answer supervision tone on the local loop when it receives “200 OK” on far end answer. This feature is for SIP only. This signal support requires the MXK-POTS-72.

For SIP, the LSAS tone or reverse battery signal are configured via the **features** parameter in the **subscriber-voice** profile. These options — **lss-tone** and **lss-rb** are mutually exclusive, so cannot be set on the same interface. These **feature** options are also mutually exclusive with **hookflash**.

Tones are defined by country as defined in system 0. The MXK provides a 16KHz tone for Thailand and 12KHz for other countries.

Once **lss-rb** or **lss-tone** is set, the subscriber must be disabled and enabled (or bounced) for the feature to take effect.

Configuring LSAS tone

To configure LSAS tone, the tone is defined by the country as configured in system 0. This feature requires the MXK-POTS-72.

1 Set/Verify the country code in system 0

```
zSH> get system 0
system 0
syscontact: -----> {Zhone Global Services and Support 7195 Oakport Street
Oakland Ca. (877) Zhone20 (946-6320) Fax (510)777-7113 support@zhone.com}
sysname: -----> {Zhone MxK}
syslocation: -----> {Oakland}
enableauthtraps: -----> {disabled}
setserialno: -----> {0}
zmsexists: -----> {false}
zmsconnectionstatus: --> {inactive}
zmsipaddress: -----> {0.0.0.0}
configsyncexists: -----> {false}
configsyncoverflow: ---> {false}
configsyncpriority: ---> {high}
configsyncaction: -----> {noaction}
configsyncfilename: ---> {}
configsyncstatus: -----> {syncinitializing}
configsyncuser: -----> {}
configsyncpasswd: -----> ** private **
numshelves: -----> {1}
shelvesarray: -----> {}
numcards: -----> {3}
ipaddress: -----> {0.0.0.0}
alternateipaddress: ---> {0.0.0.0}
countryregion: -----> {us}
primaryclocksource: ---> {0/0/0/0/0}
ringsource: -----> {internalringsourcelabel}
revertiveclocksource: -> {true}
voicebandwidthcheck: --> {false}
alarm-levels-enabled: -> {critical+major+minor+warning}
userauthmode: -----> {local}
radiusauthindex: -----> {0}
secure: -----> {disabled}
webinterface: -----> {enabled}
options: -----> {NONE(0)}
reservedVlanIdStart: --> {2000}
reservedVlanIdCount: --> {200}
```

2 Create the voice connection using the **voice add** command.

3 Update the **subscriber-voice** profile for **lss-tone**.

```
zSH> update subscriber-voice 1/4/3
subscriber-voice 1/4/3
Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}: **
read-only **
```

```

voice-endpoint1-addr-index: ---> {6}: ** read-only
**
voice-endpoint2-addr-index: ---> {5}: ** read-only
**
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}: **
read-only **
features: ----->
{hookflash+onhooksignaling+callwait}:
onhooksignaling+callwait+lss-tone
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

- 4 Bounce the voice port for the feature to take effect.

```
zSH>voice bounce 1-4-3-0/voicefxs
```

Bouncing the port disables then enables the connection, so that the added feature will take effect.

Configuring reverse battery signal

- 1 Create the voice connection using the **voice add** command.
- 2 Update the **subscriber-voice** profile for **lss-rb**.

```

zSH> update subscriber-voice 1/4/2
subscriber-voice 1/4/2
Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}: **
read-only **
voice-endpoint1-addr-index: ---> {4}: ** read-only
**
voice-endpoint2-addr-index: ---> {3}: ** read-only
**
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}: **
read-only **
features: ----->
{hookflash+onhooksignaling+callwait}:
onhooksignaling+callwait+lss-rb
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

- 3 Bounce the voice port for the feature to take effect.

```
zSH>voice bounce 1-4-2-0/voicefxs
```

Bouncing the port disables then enables the connection, so that the added feature will take effect.

DTMF mode support per port basis

DTMF(Dual Tone Multi-Frequency) describes touchtone dialing (a trademark of AT&T until 1984). RFC 2833 describes the payload format for carrying DTMF digits for gateways, end systems and what they call “RTP trunks”. The “RTP trunk” scenarios replaces the circuit switched trunk in telephone networks which may be a combination of circuit switched and RTP. DTMF signals can be carried inband or outband.

- In inband, DTMF events are send as part of the voice codec sampling. There is not special handling. When DTMF events are received in RTP stream, it will be converted to analog just like regular voice.
- In outband (RFC 2833), DTMF events are sent on a different RTP payload ID. DTMF events will be detected and converted to event packets that are compliant to RFC 2833 and sent across.

The MXK not only support the DTMF inband or DTMF outband (RFC 2833) for the whole system on a per VoIP server basis, but also support them on port basis on the MXK-POTS-72 card, VDSL+POTS combo cards, and ADSL+POTS combo cards. DTMF inband or DTMF outband is supported for SIP, MGCP, and Megaco protocols.

The behavior changes based on the settings in the **voip-server-entry** profile and the settings in the subscriber-voice profile.

The DTMF settings in the **subscriber-voice** profile takes precedence over the DTMF settings in the **voip-server-entry** profile:

- If neither dtmf-2833 or dtmf-inband are set in the **subscriber-voice** profile:

The behavior will be based on the dtmf-mode field of the **voip-server-entry** profile.

- If only dtmf-rfc2833 is set in the **subscriber-voice** profile:

The subscriber will support RFC 2833 only irrespective of what set in the dtmf-mode field of the **voip-server-entry** profile on the device and what set on the switch.

- If only dtmf-inband is set in the **subscriber-voice** profile:

The subscriber will support dtmf-inband only irrespective of what set in the dtmf-mode field of the **voip-server-entry** profile on the device and what set on the switch.

- If both dtmf-rfc2833 and dtmf-inband are set in the **subscriber-voice** profile:

Should be the same behavior as if the dtmf-mode field of the **voip-server-entry** is set to RFC-2833.

To enable DTMF mode on the device, use the **voip-server-entry** profile. This setting must match the setting on the switch. By default, rfc2833 is enabled.

```
zSH> update voip-server-entry 1/1
```

```

voip-server-entry 1/1
Please provide the following: [q]uit.
zhoneVoipServerAddrType: -----> {ipv4}:
zhoneVoipServerAddr: -----> {192.168.49.1}:
zhoneVoipServerUdpPortNumber: -----> {5060}:
zhoneVoipServerId: -----> {metaswitch}:
protocol: -----> {sip}: ** read-only **
sendCallProceedingTone: -----> {false}:
rtcpEnabled: -----> {false}:
rtcpPacketInterval: -----> {5000}:
interdigitTimeOut: -----> {10}:
ipTos: -----> {0}:
systemDomainName: -----> {}:
expires-invite-value: -----> {3600}:
expires-register-value: -----> {3600}:
expires-header-method: -----> {register}:
session-timer: -----> {off}:
session-expiration: -----> {180}:
session-min-session-expiration: ---> {180}:
session-caller-request-timer: -----> {no}:
session-callee-request-timer: -----> {no}:
session-caller-specify-refresher: -> {omit}:
session-callee-specify-refresher: -> {uac}:
dtmf-mode: -----> {rfc2833}: inband or rfc2833
rtp-termid-syntax: -----> {}:
rtpDSCP: -----> {0}:
signalingDSCP: -----> {0}:
dtmf-payload-id: -----> {101}:
register-ready-timeout: -----> {10}:
message-retry-count: -----> {1}:
voip-features: -----> {NONE(0)}:
transport-protocol: -----> {udp}:
signalling-local-port-number: -----> {5060}:
.....
Save changes? [s]ave, [c]hange or [q]uit:

```

To configure the DTMF inband (dtmf-inband) or RFC 2833 (dtmf-rfc2833) of on port basis, use the **subscriber-voice** profile. By default, RFC 2833 and DTMF mode are disabled.

```

zSH> update subscriber-voice 1/4/3
subscriber-voice 1/4/3
Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}: ** read-only **
voice-endpoint1-addr-index: ---> {6}: ** read-only **
voice-endpoint2-addr-index: ---> {5}: ** read-only **
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}: ** read-only **
features: -----> {hookflash+onhooksignaling+callwait}:
onhooksignaling+callwait+dtmf-rfc2833 dtmf-inband or dtmf-rfc2833
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Configuring DTMF mode per subscriber

To configure DTMF mode on port basis. You can set DTMF mode to dtmf-rfc2833 only, dtmf-inband only, or set them both. By default, both dtmf-rfc2833 and dtmf-inband are disabled. This feature requires the MXK-POTS-72 card, VDSL combo cards, and ADSL combo cards.

This example shows how to configure DTMF mode on a port after creating the POTs and VoIP connection:

- 1 Create the voice connection using the **voice add** command.
- 2 Update the **subscriber-voice** profile for **dtmf-rfc2833** and **dtmf-inband**.

```
zSH> update subscriber-voice 1/4/3
subscriber-voice 1/4/3
Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}: **
read-only **
voice-endpoint1-addr-index: ---> {6}: ** read-only
**
voice-endpoint2-addr-index: ---> {5}: ** read-only
**
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}: **
read-only **
features: ----->
{hookflash+onhooksignaling+callwait}:
onhooksignaling+callwait+dtmf-rfc2833+dtmf-inband
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

- 3 Bounce the voice port for the feature to take effect.

```
zSH>voice bounce 1-4-3-0/voicefxs
```

Bouncing the port disables then enables the connection, so that the added feature will take effect.

Data exchange only

The MXK allows only data to be exchanged for the entire duration of the VoIP call. You can use this feature for fax or dial-up modem, etc. It makes the data line or fax line more reliable. The dataonly feature can be enabled during the creation of a subscriber or by modifying a **subscriber-voice** profile.

This feature works on all MXK voice cards, and all voice protocols (e.g MGCP, SIP, etc.).

Note that the dataonly feature and voiceonly feature are mutually exclusive.

Configuring data only per subscriber

This example shows two methods to configure data only feature on a port:

- 1 Method 1: To enable dataonly feature when creating voice connection:

```
zSH> voice add pots 1-7-5-0/voicefxs voip ethernet2/ip dn 201200614 name aaln/S1/9
+feature hookflash+onhooksignaling+dataonly
Created subscriber-voice 1/7/5
Created subscriber-voice-pots 9
Created subscriber-voice-voip 10
```

- 2 Method 2: To enable dataonly feature on an existing voice connection:

- a Modify the **subscriber-voice** profile. By default, dataonly is disabled.

```
zSH> update subscriber-voice 1/4/5
subscriber-voice 1/4/5
Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}: ** read-only **
voice-endpoint1-addr-index: ---> {10}: ** read-only **
voice-endpoint2-addr-index: ---> {9}: ** read-only **
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}: ** read-only **
features: -----> {hookflash+onhooksignaling}:
hookflash+onhooksignaling+dataonly
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

- b Bounce the voice port for the feature to take effect.

```
zSH>voice bounce 1-4-5-0/voicefxs
```

Voice exchange only

The MXK allows only voice codecs to be exchanged for the entire duration of the VoIP call, ignores any fax, data, dial-up modem tones coming from the line. It makes the voice line more reliable. The voiceonly feature can be enabled during the creation of a subscriber or by modifying a **subscriber-voice** profile.

This feature works on all MXK voice cards, and all voice protocols (e.g MGCP, SIP, etc.).

Note that the dataonly feature and voiceonly feature are mutually exclusive.

Configuring voice only per subscriber

This example shows two methods to configure voice only feature on a port:

- 1 Method 1: To enable voiceonly feature when creating voice connection:

```
zSH> voice add pots 1-7-5-0/voicefxs voip ethernet2/ip dn 201200614 name aaln/S1/9
+feature hookflash+onhooksignaling+voiceonly
```

```
Created subscriber-voice 1/7/5
Created subscriber-voice-pots 9
Created subscriber-voice-voip 10
```

2 Method 2: To enable voiceonly feature on an existing voice connection:

- a** Modify the **subscriber-voice** profile. By default, voiceonly is disabled.

```
zSH> update subscriber-voice 1/4/5
subscriber-voice 1/4/5
Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}: ** read-only **
voice-endpoint1-addr-index: ---> {10}: ** read-only **
voice-endpoint2-addr-index: ---> {9}: ** read-only **
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}: ** read-only **
features: -----> {hookflash+onhooksignaling}:
hookflash+onhooksignaling+voiceonly
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

- b** Bounce the voice port for the feature to take effect.

```
zSH>voice bounce 1-4-5-0/voicefxs
```

Plar

SIP (Session Initiation Protocol) negotiates features when the session is established. PLAR (Private Line Automatic Ringdown) means that two lines are connected, so that when a phone on one end of the line goes off hook, the phone on the other end rings. A SIP PLAR server entry is automatically created when user specifying the **voice add** command with the **plar** option.

For details about how to configure SIP PLAR, refer to [SIP PLAR on page 500](#).

Configuring SIP PLAR per subscriber

This example shows two methods to configure voice only feature on a port:

1 Method 1: To enable SIP PLAR feature when creating voice connection:

```
zSH> voice add pots 1-1-13-0/voicefxs voip ethernet2/ip dn 201200614 name 7770001
plar 10.6.20.2 reg 0 sub 7770001 enable
mdlog _VoiceGetPlarIpAddrFromCli : plar ip addr 10.6.20.2
Created subscriber-voice 1/4/23
Created subscriber-voice-pots 967
mdlog _plar 10.6.20.2Created subscriber-voice-voip 968
Interface 1-1-13-0/voicefxs's admin status is set to ENABLED
```

2 Method 2: To enable SIP PLAR feature on an existing voice connection:

- a** Modify the **subscriber-voice** profile. By default, PLAR is disabled.

```

zSH> update subscriber-voice 1/4/5
subscriber-voice 1/4/5
Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}: ** read-only **
voice-endpoint1-addr-index: ---> {10}: ** read-only **
voice-endpoint2-addr-index: ---> {9}: ** read-only **
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}: ** read-only **
features: -----> {hookflash+onhooksignaling}:
hookflash+onhooksignaling+plar
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

- b** Bounce the voice port for the feature to take effect.

```
zSH>voice bounce 1-4-5-0/voicefxs
```

Hotline and Warmline

When hot line is enabled, the phone will immediately dial the hot line number when the phone is taken off hook. Hot line is primarily used for Hotel “house line” applications where phones throughout the hotel/motel can only be used to call the front desk.

Warm line is a variation of hot line, but with a configurable timer value for dialing. When a user goes off-hook, if no digits are dialed before the warm line timer expires, then a call will go out the configured hot line number. If one or more digits is dialed before the warm line timer expires, then the warm line feature is disabled and the line operates in normal mode until it goes back on-hook.

Configuring hot line or warm line mode per subscriber

You can set line mode to hot line or warm line. By default, both hot line and warm line are disabled. Note that hot line and warm line features cannot be enabled at the same time.

This example shows how to configure hot line mode on a port after creating the POTs and VoIP connection:

- 1** Create the voice connection using the **voice add** command.
- 2** Update the **subscriber-voice** profile for **hotline**.

```

zSH> update subscriber-voice 1/4/3
subscriber-voice 1/4/3
Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}: **
read-only **
voice-endpoint1-addr-index: ---> {6}: ** read-only
**

```

```

voice-endpoint2-addr-index: ---> {5}: ** read-only
**
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}: **
read-only **
features: ----->
{hookflash+onhooksignaling+callwait}:
onhooksignaling+callwait+hotline
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

- 3 Bounce the voice port for the feature to take effect.

```
zSH>voice bounce 1-4-3-0/voicefxs
```

Bouncing the port disables then enables the connection, so that the added feature will take effect.

Cut-off on Disconnect

CoD (Cut-off on Disconnect) sends a signal on the on-hook event which tells the softswitch that the phone has been hung up.

Configuring CoD per subscriber

You can set CoD mode per port base. By default, CoD is disabled.

This example shows how to configure CoD mode on a port after creating the POTs and VoIP connection:

- 1 Create the voice connection using the **voice add** command.
- 2 Update the **subscriber-voice** profile for **CoD**

```

zSH> update subscriber-voice 1/4/3
subscriber-voice 1/4/3
Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}: **
read-only **
voice-endpoint1-addr-index: ---> {6}: ** read-only
**
voice-endpoint2-addr-index: ---> {5}: ** read-only
**
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}: **
read-only **
features: ----->
{hookflash+onhooksignaling+callwait}:
onhooksignaling+callwait+cod
.....

```

```
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

- 3 Bounce the voice port for the feature to take effect.

```
zSH>voice bounce 1-4-3-0/voicefxs
```

Bouncing the port disables then enables the connection, so that the added feature will take effect.

Always off hook

With this feature, system assumes the subscriber goes offhook as soon as the incoming call is setup irrespective of the hookstate of the subscriber. That way RTP stream is established right away.

Configuring alwaysoffhook per subscriber

You can set alwaysoffhook mode per port base. By default, alwaysoffhook is disabled.

This example shows how to configure alwaysoffhook mode on a port after creating the POTs and VoIP connection:

- 1 Create the voice connection using the **voice add** command.
- 2 Update the **subscriber-voice** profile for **alwaysoffhook**:

```
zSH> update subscriber-voice 1/4/3
subscriber-voice 1/4/3
Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}: **
read-only **
voice-endpoint1-addr-index: ---> {6}: ** read-only
**
voice-endpoint2-addr-index: ---> {5}: ** read-only
**
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}: **
read-only **
features: ----->
{hookflash+onhooksignaling+callwait}:
onhooksignaling+callwait+alwaysoffhook
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

- 3 Bounce the voice port for the feature to take effect.

```
zSH>voice bounce 1-4-3-0/voicefxs
```

Bouncing the port disables then enables the connection, so that the added feature will take effect.

Centrex

The difference between a centrex subscriber and a non-centrex subscriber is when the subscriber goes onhook after initiating a local three way conference.

- A centrex subscriber initiates a call transfer between the remaining subscribers in the conference. That way the other two parties can still be in a conversation.
- A non-centrex subscriber disconnects the conference completely and all the parties involved in the conference are disconnected.

Configuring Centrex per subscriber

You can set Centrex per port base. By default, centrex is disabled.

This example shows how to configure centrex mode on a port after creating the POTs and VoIP connection:

- 1 Create the voice connection using the **voice add** command.
- 2 Update the **subscriber-voice** profile for **centrex**:

```
zSH> update subscriber-voice 1/4/3
subscriber-voice 1/4/3
Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}: **
read-only **
voice-endpoint1-addr-index: ---> {6}: ** read-only
**
voice-endpoint2-addr-index: ---> {5}: ** read-only
**
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}: **
read-only **
features: ----->
{hookflash+onhooksignaling+callwait}:
onhooksignaling+callwait+centrex
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

- 3 Bounce the voice port for the feature to take effect.

```
zSH>voice bounce 1-4-3-0/voicefxs
```

Bouncing the port disables then enables the connection, so that the added feature will take effect.

Advanced features

- [ESA, page 542](#)
- [ToS configuration for voice signaling packet, page 542](#)
- [T.38 fax, page 544](#)

ESA

For SIP, SIP PLAR, or H.248 voice connections, the MXK provides emergency calling services during network or equipment failures that cause a loss of connection to the configured SIP/SIP PLAR/ H.248 server or voice gateway MALC.

If the MXK loses SIP/SIP PLAR /H.248 communication with the softswitch, the MXK will continue to process calls locally between subscribers in the same MXK chassis to another reachable MXKs in the ESA cluster. POTS subscribers on the same MXK can make calls (voice, fax, modem) between each other as well as calls to other reachable MXKs in the ESA cluster, based on the predefined dial plans for each MXK in the ESA cluster.

Refer to the following sections for the detail configuration:

[Emergency Stand Alone \(ESA\) for SIP, page 489](#)

[ESA for H.248, page 514](#)

ToS configuration for voice signaling packet

ToS for voice signaling packets is set in the **voip-server-entry** profile.

[Table 44](#) specifies the IP ToS settings used in the **voip-server-entry** profile based on IP Precedence bits.



Note: When setting ToS for IP packets in the **ip-interface-record** profile, the values in the precedence bits column are used, when setting ToS for voice signaling packets in the **voip-server-entry** profile, the values in the ToS value column are used.

Table 44: IP ToS settings and IP Precedence bits

Precedence bits	ToS value
0 (Routine)	0
1 (Priority)	32
2 (Immediate)	64
3 (Flash)	96
4 (Flash override)	128

Table 44: IP ToS settings and IP Precedence bits (Continued)

Precedence bits	ToS value
5 (CRITIC/ECP.)	160
6 (Internetwork control)	192
7 (Network control)	224

Configuring VoIP QoS

To add ToS to voice signaling packets, you must configure the **ipTos** parameter of the **voip-server-entry** profile.

- 1 View the existing **voip-server-entry** profiles if necessary.

```
zSH> list voip-server-entry
voip-server-entry 1/1
1 entry found.
```

- 2 Configure the **ipTos** parameter with the ToS value (see [Table 44](#)) in the **voip-server-entry** profile to add the ToS value to the signaling voice packets.

```
zSH> update voip-server-entry 1/1
voip-server-entry 1/1
Please provide the following: [q]uit.
zhoneVoipServerAddrType: -----> {ipv4}:
zhoneVoipServerAddr: -----> {172.16.160.3}:
zhoneVoipServerUdpPortNumber: -----> {5060}:
zhoneVoipServerId: -----> {generic}:
protocol: -----> {sip}: ** read-only **
sendCallProceedingTone: -----> {false}:
rtcpEnabled: -----> {false}:
rtcpPacketInterval: -----> {5000}:
interdigitTimeout: -----> {10}:
ipTos: -----> {0}: 160
systemDomainName: -----> {}:
expires-invite-value: -----> {3600}:
expires-register-value: -----> {3600}:
expires-header-method: -----> {register}:
session-timer: -----> {off}:
session-expiration: -----> {180}:
session-min-session-expiration: ----> {180}:
session-caller-request-timer: -----> {no}:
session-callee-request-timer: -----> {no}:
session-caller-specify-refresher: -> {omit}:
session-callee-specify-refresher: -> {uac}:
dtmf-mode: -----> {rfc2833}:
rtp-termid-syntax: -----> {}:
rtpDSCP:-----> (0)
signalingDSCP:-----> (0)
dtmf-payload-id: -----> {101}:
register-ready-timeout: -----> {10}:
message-retry-count: -----> {1}:
```

```

voip-features: -----> {NONE(0)}:
transport-protocol: -----> {udp}:
signalling-local-port-number: -----> {5060}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

T.38 fax

T.38 fax service enables fax messages to be transported across VoIP networks between G3 fax terminals. When configured for SIP or SIP PLAR and T.38, MXK provides a T.38 fax relay service between two devices configured for the same VoIP protocol. If one side of the T.38 connection is not configured for T.38 support, the fax call reverts to g.711 pass through when this option is configured. Otherwise, the fax may not go through.

By default, T.38 fax service is disabled.

This section contains the following procedures;

- [T.38 to VoIP connection](#)
- [T.38 fax to Voice Gateway V5.2/GR303 connection with SIP PLAR](#)
- [Route T.38 fax between MXKs with Voice Gateway](#)



Note: The T.38 fax service can also be configured on VoIP connections using the voicegateway card.



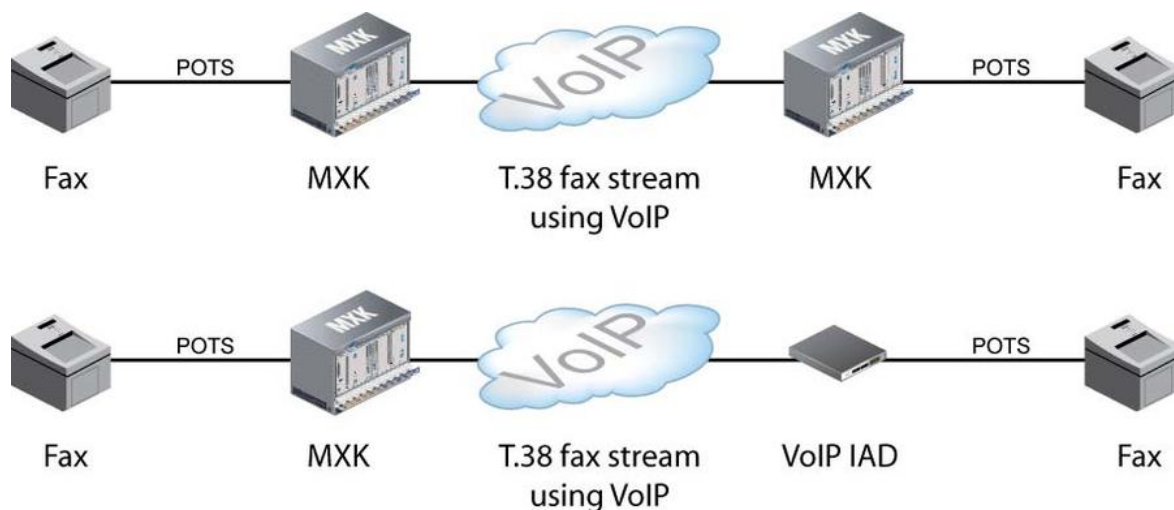
Note: When using T.38 fax, be sure that all the devices on the network which are involved in the T.38 transmission/reception are correctly configured for T.38 fax service.

T.38 to VoIP connection

The MXK supports T.38 fax streams across a VoIP network. The MXK can be connected to another MXK or a VoIP IAD device.

[Figure 72](#) illustrates the T.38 fax streams using VoIP between MXK devices, and between a MXK and a VoIP IAD configured for T.38.

Figure 72: T.38 between MXK devices or VoIP IAD



Configuring T.38 fax service when creating a POTS-to-VoIP connection

The MXK supports T.38 service options for either **t38udptl** or **t38none**. The **t38udptl** option enables T.38 service using UDP IP packets. The **t38none** option disables the service.



Note: The **t38rtp** option is currently not supported.

To enable T.38 fax service when creating a VoIP connection

Specify the T.38 option when configuring a voice call with the **voice add** command for the POTS and VoIP connections. The **subscriber-voice-voip profile** settings are updated based on the command options.

If configure T.38 fax for SIP connection, use this example:

```
voice add pots 1-10-1-0/voicefxs voip ethernet2-100/ip dn 5105330203 name
5105330203 t38fax t38udptl reg 1 enable
```

If configure T.38 fax for MGCP connection, use this example:

```
voice add pots 1-10-1-0/voicefxs voip ethernet2-100/ip dn 201202999 name
aaln/1 t38fax t38undptl reg 1 enable
```

If configure T.38 fax for H.248 connection, use this example:

```
voice add pots 1-10-1-0/voicefxs voip ethernet2-100/ip dn 201202999 name
tp/0000 t38fax t38undptl reg 1 enable
```

Configuring T.38 fax service after creating a POTS-to-VoIP connection

If a POTS-to-VoIP connection is already created for SIP, MGCP, or H.248, you can update the **subscriber-voice-voip profile** to enable the T.38 fax service. After updating the subscriber-voice-voip profile, the voice subscriber must be disabled and then re-enabled for the changes to be effective.

- 1 List the subscriber voice profiles.

```
zSH> list subscriber-voice
subscriber-voice 1/2/26
subscriber-voice 1/2/27
2 entries found.
```

- 2 Use the **get subscriber-voice** command to find the voice-endpoint1-addr-index, which matches the subscriber-voice-voip profile index.

```
zSH> get subscriber-voice 1/2/26
Please provide the following: [q]uit.
voice-connection-type: -----> {voiptopots}:
voice-endpoint1-addr-index: ---> {52}:
voice-endpoint2-addr-index: ---> {51}:
voice-connection-description: -> {}:
voice-admin-status: -----> {enabled}:
huntgroup: -----> {false}:
features: -----> {hookflash+onhooksignaling+callwait}:
```

- 3 Enable the t38 fax in the subscriber voice voip profile.

```
zSH> update subscriber-voice-voip 52 (the endpoint1-addr-index in subscriber-voice profile.)
Please provide the following: [q]uit.
voip-username: -----> {9990002}:
directory-number: -----> {9990002}:
ip-interface-index: -----> {ethernet2-2/ip}:
preferred-codec: -----> {g729a}:
g711-fallback: -----> {true}:
frames-per-packet: -----> {4}:
g726-byte-order: -----> {bigendian}:
voip-password: -----> {}:
voip-plar: -----> {false}:** read-only **
voip-plar-dest-ipaddrtype: -> {ipv4}:
voip-plar-dest-ipaddr: -----> {}:
voip-plar-udp-port: -----> {5060}:
registration-server: -----> {0}:
t38-fax: -----> {t38none}:t38udpt1
voip-authuser: -----> {36}:
hotline-directory-number: --> {36}:
hotline-initial-timer: -----> {0-0}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
Record created.
```

- 4 Use **voice bounce** command to make the change to be effective.

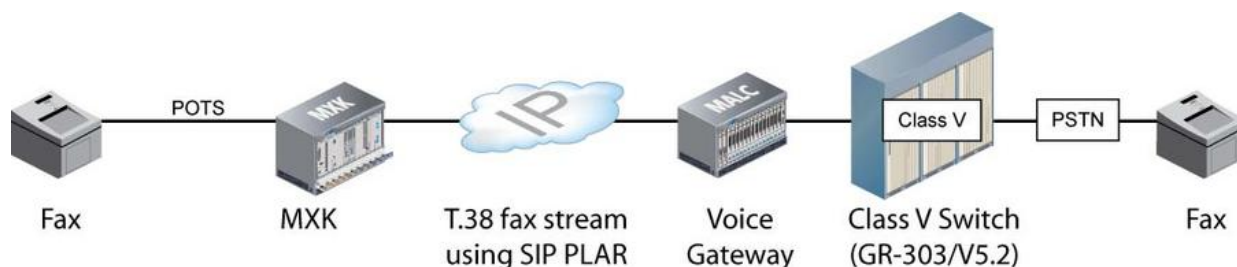
```
zSH> voice bounce 1/2/26
```

T.38 fax to Voice Gateway V5.2/GR303 connection with SIP PLAR

The MXK supports T.38 fax streams across a VoIP network using SIP PLAR. In this configuration, the fax signal is sent to the MALC with a voicegateway card, and then forwarded to the PSTN as either a GR-303 or V5.2 fax signal.

Figure 73 illustrates the T.38 fax stream using SIP PLAR between MXK and MALC with a voicegateway card connected to a class V switch and the PSTN.

Figure 73: SIP PLAR T.38 between MXK and MALC Voicegateway to PSTN



Creating T.38 fax to VG connections with SIP-PLAR

The MXK supports T.38 fax streams across a VoIP network using SIP PLAR. In this configuration, one MXK converts the POTS signal to VoIP and sends the T.38 fax signal across the VoIP network. A MALC with the voicegateway card receives the T.38 signal and sends it to the Class V switch for processing across the PSTN.

- 1 On the MXK converting the POTS to VOIP signal, specify the T.38 option when configuring a voice call with the **voice add** command for the POTS and SIP connections. 199.190.212.238 is the VG MALC's IP address. The **subscriber-voice-voip profile** settings are updated based on the command options.

```
voice add pots 1-5-3-0/voicefxs voip ethernet1/ip dn 7360001 name 7360001
plar 199.190.212.238 codec g729a t38fax t38udpt1 reg 0 sub 7360001 enable
```

- 2 On the MALC with the voicegateway card, use the **voice add** command to configure the connection for either VoIP to GR303 or VoIP to V5.2.

For the configuration refer to the *MALC Configuration Guide*.

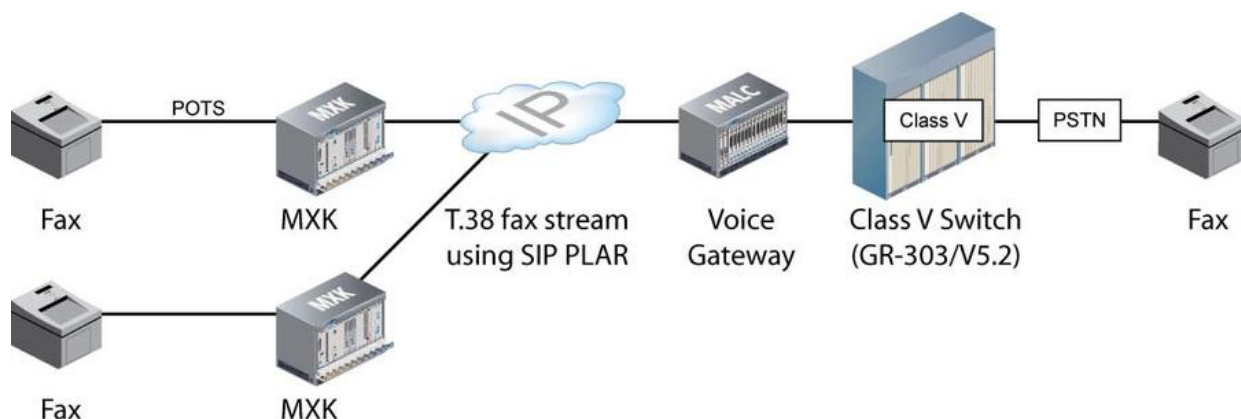
Route T.38 fax between MXKs with Voice Gateway

The MXK supports T.38 fax streams across a VoIP network using SIP PLAR to another MXK device in the network. In this configuration, the fax signal is sent to the MALC with the voicegateway card, and then forwarded to the

Class V switch, which routes the call back through the VoIP network to another MXK.

Figure 74 illustrates the T.38 fax stream using SIP PLAR between a MXK connected to a MALC with the voicegateway card. When the signal reaches the MALC with the voicegateway card, the Class V switch routes the signal to another MXK in the VoIP network to process the POTS fax.

Figure 74: SIP PLAR T.38 between MXK and MALC Voicegateway to POTS fax



Configuring T.38 using VG to route POTS fax

- 1 On the MXK devices converting the POTS to VOIP signal, specify the T.38 option when configuring a voice call with the **voice add** command for the POTS and voice connections. The **subscriber-voice-voip profile** settings are updated based on the command options.

Feeder MXK 1:

```
voice add pots 1-1-1-0/voicefxs voip ethernet3/ip dn 7360001 name 7360001
plar 199.190.212.238 t38fax t38udpt1 reg 0 sub 7360001 enable
```

Feeder MXK 2:

```
voice add pots 1-1-1-0/voicefxs voip ethernet3/ip dn 7360002 name 7360002
plar 199.190.212.238 t38fax t38udpt1 reg 0 sub 7360002 enable
```

- 2 On the MALC with the voicegateway card, use the **voice add** command to configure the T.38 connection for VoIP to GR303 or VoIP to V5.2.

For the configuration refer to the *MALC Configuration Guide*.

7

MXK PSEUDO WIRE EMULATION (PWE) CONFIGURATION

This chapter describes the Pseudo Wire Emulation capabilities and configuration on the MXK. This chapter includes:

- [PWE on the MXK, page 549](#)
- [Creating PWE connections, page 562](#)
- [PWE alarms, logs and traps, page 570](#)
- [PWE operational status, page 573](#)
- [PWE commands, page 576](#)

PWE on the MXK

This section contains the following sections which describe PWE on the MXK.

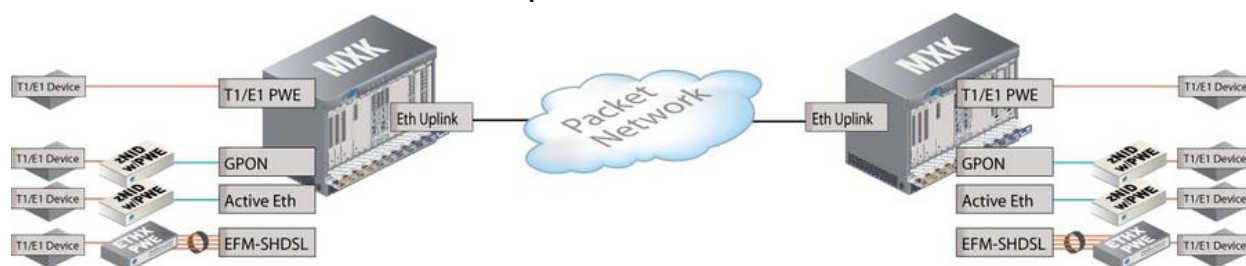
- [Overview, page 550](#)
- [PWE connections, page 552](#)
- [The pwe-tdm add command, page 556](#)

Overview

PWE provides a means for alternative carriers to offer legacy services. Incumbent carriers can migrate legacy service to packet switching networks to remain competitive.

Zhone provides PWE solutions both from line cards on the MXK with individual ports on the T1/E1 PWE line card or from connecting to other Zhone products. Other Zhone products such as GPON and Active Ethernet zNID ONTs and EtherXtend CPE devices offer T1/E1 ports for PWE connections.

Figure 75: Zhone offers PWE connection directly from the MXK or downstream on other Zhone products



In order to transmit T1/E1 connections across a packet network the information is packetized and encapsulated to be transferred transparently to the T1/E1 device on the other end as shown in [Figure 75](#).

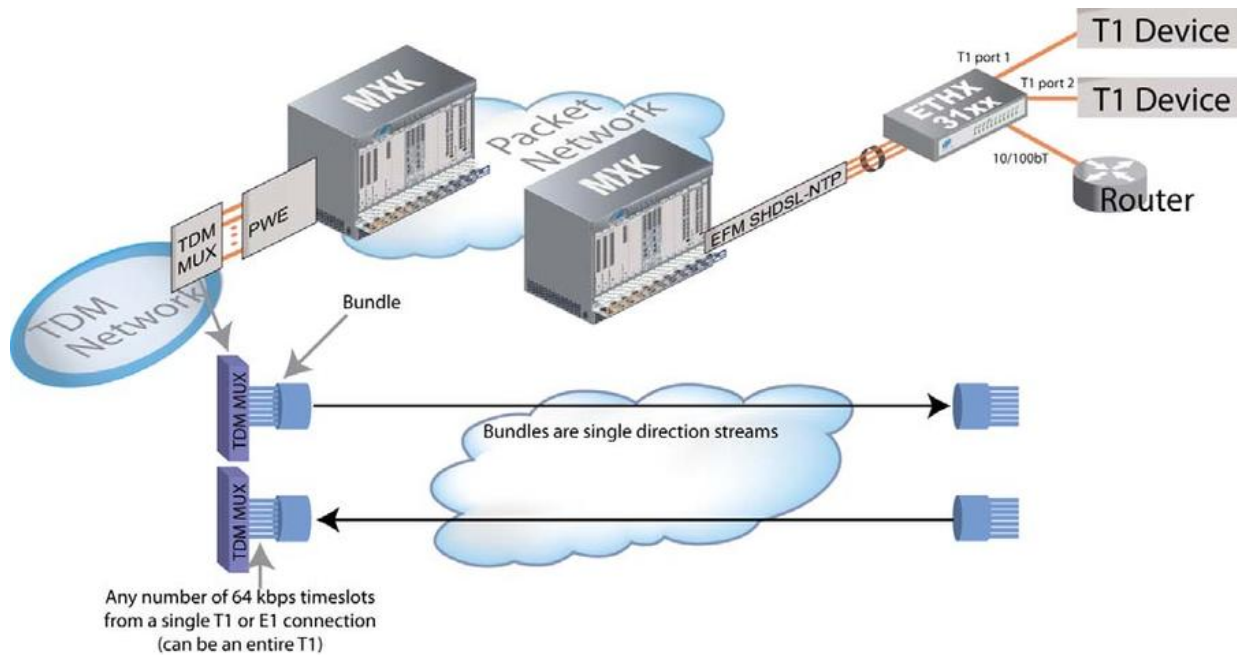
This chapter shows how to configure PWE connections based on T1/E1 ports on MXK T1/E1 line cards.

Figure 76: The configuration section of this chapter describes PWE connections which have IP termination on the MXK, not on other devices.



For using PWE, an IP based solution, with EAPS, a layer 2 bridging solution, see [PWE solution with EAPS on page 561](#).

Figure 77: Since bundles are single direction, there is an opposite bundle for full duplex communications

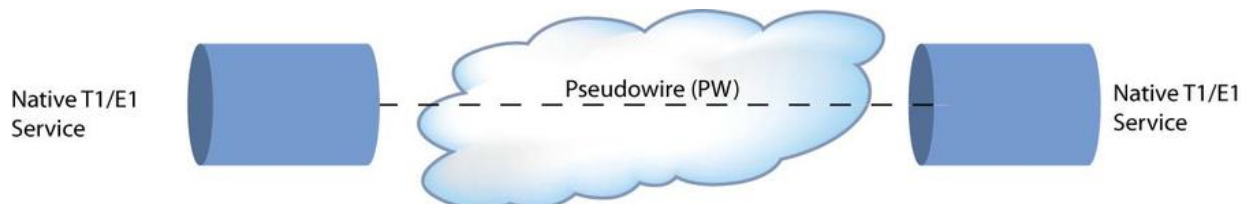


As long as an IP route can be created between the source PWE access device and the remote PWE access device, whether it be an Internet cloud or an EFM bonded group as shown in some of the examples, a PWE connection can be made using the **pwe-tdm add** command.

PWE connections

PWE uses bundles, streams of bits which have originated from the same physical interface which are transmitted to a destination device. Bundles may be made up of any number of 64kbps timeslots originating from a single T1 or E1 and may go up to an entire T1/E1. Bundles are single direction streams. Often there is a reciprocal bundle going in the other direction for full duplex communication between both ends of the pseudowire as shown in [Figure 78](#).

Figure 78: Pseudowire (PW)



We use the **pwe-tdm add** command to create one way connections. Both ends of the connection must be configured for traffic to pass. When both ends are configured it creates a full duplex connection.

Zhone's PWE solutions set up both ends of the pseudowire. When you use the **pwe-tdm add** command to set up a connection with a source and destination, it not only sets up the source to send, but also to receive frames; likewise on the remote device.

PWE timing

The proper delivery of packets requires a clocking mechanism. The configuration procedure becomes more complex when you overlay one of the PWE timing recovery modes and, where applicable, the external clock sources.

Latency with voice and data services

Latency is the amount of time from the source sending a packet to the destination receiving the packet. The MXK T1/E1 PWE solution can operate without error even when several seconds of latency exist from one end of the pseudo-wire to the other.

The network on which the pseudo-wire is operating should be engineered to take into account the services being offered. Services such as Voice PRI would not work properly in a high latency network. A T1/E1 circuit used only for data transfer may be able to tolerate a high level of latency.



Note: The limiting factor of network latency is dependent on the service offered and not on the capability of the MXK PWE.

Some applications can tolerate higher latency than others. The primary source of latency in a PWE connection is the Jitter Buffer that is necessary to

compensate for all of the packet delay variation that has been introduced by the network itself. To reduce latency, it is necessary to ensure that all PWE packets are handled with expedited priority through the network. When the network handles all PWE traffic as high priority packets, packet delay variation will be reduced, and a smaller Jitter Buffer can be used. As a result, end to end latency will be reduced

Network jitter can have a negative affect on a circuit emulation over packet service. The T1/E1 circuit must be played out a constant rate to successfully emulate the circuit. The MXK PWE line card implements a buffering scheme to dampen the affect of jitter in the network, but buffering will not help if jitter is too pronounced. If the packet inter arrival rate is too large the playout buffer will starve and the user equipment will lose framing. If the packet inter arrival rate is too short for a time period the playout buffer could overflow causing packet loss. Acceptable jitter will vary depending on the size of packets and the size of the buffer, but a good recommendation is to keep jitter under 2ms.

It is important that PWE traffic in the network be classified and treated as high priority, low latency traffic. PWE traffic will normally be a lower priority than management traffic and a higher priority than VoIP traffic.

PWE timing recovery modes

The PWE timing recovery mode defines how clocking is provided to match up the packet arrival times in the process of fragmenting and reconstructing the data stream.

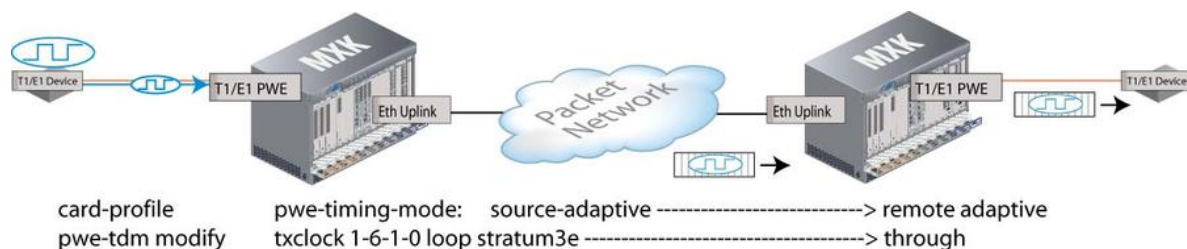


Note: Notice that each PWE card may only support one timing recovery mode for all the ports on that card.

For more information on the MXK's clocking options, please see [Chapter 3, MXK Clocking, on page 173](#).

Zhone supports three combinations of PWE timing modes:

- Timing is provided from the TDM source through packet to destination



The timing comes from the TDM source and is encapsulated in the packet transmitted to the destination.

Source:

- **pwe-timing-mode: source-adaptive** (in card profile)
- pwe-tdm modify txclock 1-6-1-0 **loop** stratum3e

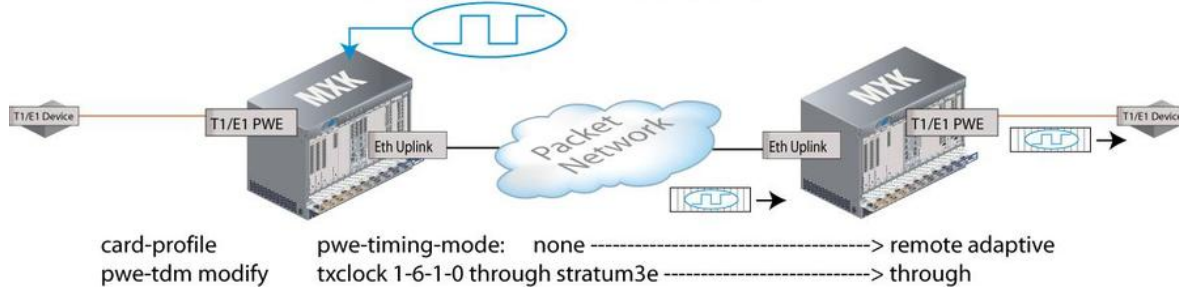
loop means the **txclock** is coming from the link.

Destination:

- **pwe-timing-mode: remote-adaptive** (in card profile)
- pwe-tdm modify txclock 1-6-1-0 **through** stratum3e
- Timing provided to source MXK from an external clock through packet to destination

for backplane clock source:

update system-clock-profile system-clock-eligibility = true 1-10-1-0/ds1
(example assumes a ds1 port receiving clock)



The timing comes from an external clock source and is encapsulated in the packet transmitted to the destination

Source:

- **pwe-timing-mode: none** (in card profile)
- pwe-tdm modify txclock 1-6-1-0 **through** stratum3e

through means the **txclock** is coming from the backplane (and in these examples the backplane is getting it from an external source)

- for clocking to backplane (the source is received through another ds1 port) update system-clock-profile system-clock-eligibility = true 1-10-1-0/ds1

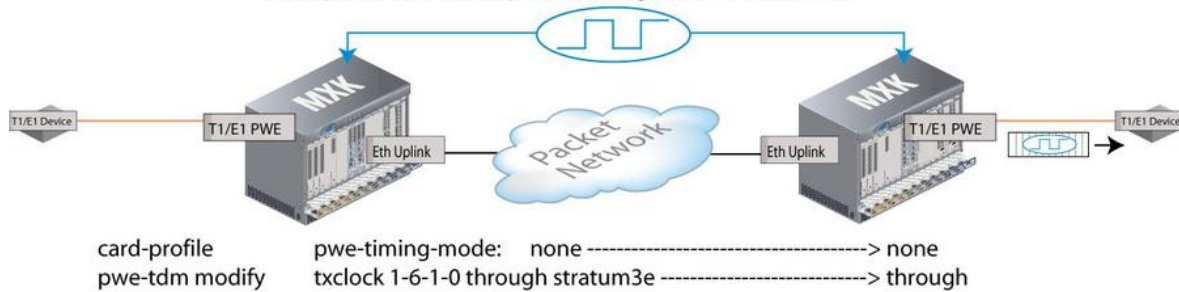
Destination:

- **pwe-timing-mode: remote-adaptive** (in card profile)
- pwe-tdm modify txclock 1-6-1-0 **through** stratum3e

- The same external clock provided to both source and destination

for backplane clock source:

update system-clock-profile system-clock-eligibility = true 1-10-1-0/ds1
(example assumes a ds1 port receiving clock on each MXK)



The timing for each MXK comes from the same external clock source.

Source:

- **pwe-timing-mode: none** (in card profile)
- pwe-tdm modify txclock 1-6-1-0 **through** stratum3e
- for clocking to backplane (the source is received through another ds1 port) update system-clock-profile system-clock-eligibility = true 1-10-1-0/ds1

Destination:

- **pwe-timing-mode: none** (in card profile)
- pwe-tdm modify txclock 1-6-1-0 **through** stratum3e
- for clocking to backplane (the source is received through another ds1 port) update system-clock-profile system-clock-eligibility = true 1-10-1-0/ds1

Since PWE connections are defined as one way with a source and a destination, these connections may use the same type of timing for the other direction.

The pwe-tdm add command

The PWE connection is created by the **pwe-tdm add** command, however as we have seen in the [PWE connections](#) section, there are other items, such as defining the pwe-timing-mode and matching it to the configuration. There are also other items which must be configured to match the configuration scenario. The line must be set for T1/E1 and further T1/E1 options, zero suppression, getting clock sources into the MXK and setting up IP addresses. All of these issues are described in [Creating PWE connections on page 562](#).

The **pwe-tdm add** command sets up the PWE end points, priority, channels (if using structured), payload size and jitter buffer, patterns which are used for overflow or underflow of the jitter buffer, and whether the connection is to be used for ISDN.

The parts of the pwe-tdm add command

- Define the port: **<shelf-slot-port-subport/ds1>**; common method for Zhone interfaces
- Identify the source and destination IP addresses and UDP ports
See [PWE IP addresses and UDP ports on page 557](#).
- Set priority: tos
PWE streams are prioritized using an IP Type of Service (tos) mechanism. Values may be from 0 to 255 with the higher number having the higher priority.
- Channelization CESoP or SAToP
Zhone's PWE solution supports structured or unstructured Circuit Emulation Service (CES) See [Channelization: SAToP and CESoP on page 558](#).
- Connection name
Usually the name will be a descriptive string to identify the connection.
- Payload size, jittermean and pattern filling for overflow/underflow of the jitter buffer
See [PWE timing on page 552](#)
- ISDN
The ISDN parameter configures whether the connection is for ISDN. See [Configuring PWE for E1 ISDN PRI on page 566](#). Zhone offers other solutions for ISDN phone services. See <XREF to voice config chapter> for more information.

PWE IP addresses and UDP ports

When configuring source IP addresses for PWE bundles on a MXK system, consideration must be given to the fact that there must be one or more IP interfaces configured on the MXK port(s) and each of these IP interfaces must have a unique IP address for proper networking and routing. Furthermore, each source IP address used for each PWE bundle in a MXK chassis must exactly match the IP address of one of the MXK IP interfaces. This match ensures that the MXK can properly route the PWE bundle from the PWE card to one of the MXK uplink ports. Additionally, by ensuring that the source IP address of each bundle exactly matches one IP interface on the MXK ports, you will be assured that the PWE bundle can be uniquely identified across an entire network.



Note: PWE connections which require IP termination on the MXK (such as the PWE ports from T1/E1 PWE card) use IPv4 for IP addressing.

UDP ports for PWE must be selected from the range of [56251..60100]. .

When configuring source UDP ports for PWE bundles on one or more MXK systems in a network, you must ensure that the pairing of the source IP address plus the source UDP port of each PWE bundle is unique across all MXKs in a network. Additionally, the source UDP port of all PWE bundles within a single MXK chassis must also be unique. Since the IP address of each MXK within a network must be unique for proper IP network design, this means that you can re-use source UDP port values in different MXK chassis. However, the source UDP ports for all bundles within a single chassis must be uniquely assigned.

When configuring UDP destination ports, the reserved IANA UDP port value of 2142 is treated as a special case and imposes additional rules on the selection of source UDP port values. When a UDP destination port value of 2142 is used on a MXK PWE bundle, the UDP source port [selected from the range 56251..60100] must be the same on both ends of the PWE connection. For example, when configuring a PWE bundle from one MXK chassis to another MXK chassis, if you use a UDP source port of 59001 and a destination UDP port of 2142 on one of the PWE cards, you must also configure the bundle on the other PWE card to have the same source and destination UDP port values of 59001 and 2142 respectively.



Note: UDP destination port 2142 must be used when connecting the MXK PWE card (mxlc24t1e1pwe.bin) to an EtherXtend 31xx.



Note: Configuring SAToP or CESoP bundles from port-to-port on the same card or port-to-port on different cards in the same chassis is supported, however, you must select the UDP destination port from the range 56251...60100. For this case you cannot use the special case UDP Destination port value of 2142.

Channelization: SAToP and CESoP

The Zhone PWE solution supports structured and unstructured CES:

- SAToP (Structure-Agnostic Time Division Multiplexing over Packet)
Used for unstructured CES
- CESoP (Circuit Emulation Service over Packet)
Used for structured (channelized) CES

The MXK passes the packets through regardless of whether the incoming information is channelized or unchannelized. In unstructured emulation (also known as unchannelized or clear channel emulation) the entire services bandwidth is emulated and reproduced at the target port. Structured emulation service (also called channelized emulation) emulates a point-to-point fractional T1/E1 (less than a full T1/E1 line), and the frame structure is maintained. Individual streams are visible and are byte aligned. This structured, channelized approach allows the T1/E1 trunks using the structured emulation service to break into multiple DS0 channels towards different destinations. See [Configuring CESoP channels on page 558](#) for more information.

The Zhone solution encapsulates the PWE frames in UDP packets for transmission over IP.

CESoP packetization

Circuit Emulation Services-over-Packet (CESoP) enhances SAToP mode transport functionality to allow the transport of structured, n x 64 kbps DS0 channels. In this way, fractional T1/E1 or individual voice channels / bundles can be transported much more efficiently over PWE by avoiding the need to transport an entire T1/E1 of bandwidth when only a few channels are required.

For a full example of configuring CESoP channels, please see [PWE with CESoP channelization, page 563](#).

Configuring CESoP channels. To configure CESoP structured mode use the **channels** option in the **pwe-tdm** command. If the channels option is not specified, the PWE will operate in SAToP unstructured mode by default.

The following shows setting up for channels in the **pwe-tdm add** command:

```
zSH> pwe-tdm add 1-7-1-0/ds1 srcip 192.168.3.1 srcudp
57001 destip 192.168.3.2 destudp 2142 tos 7 channels
1+2+3+4 payload 188 jittermean 5000 isdn disabled
Created 1-7-1-0-ds1-1/ds0bundle
```

To verify channelization use the **pwe-tdm show entry** command for the interface:

```
zSH> pwe-tdm show entry 1-7-1-0-ds1-1/ds0bundle
Pw Entry Config for PW 1-7-1-0-ds1-1/ds0bundle
```

```

type: -----> {basicCesPsn}
destip: -----> {192.168.3.2}
destudp: -----> {2142}
srcip: -----> {192.168.3.1}
srcudp: -----> {57001}
desc: -----> {}
channels: -----> {1+2+3+4}
tos: -----> {7}
adminstat: -----> {up}
payload: -----> {188}
jittermean: ----> {5000}
replacepolicy: -> {allones}
pattern: -----> {255}
isdn: -----> {disabled}

```

See [PWE with CESoP channelization, page 563](#) for a complete example.



Note: When CESoP PWE bundles are created, the **ifName** in the **if-translate** profile should not be modified. The **ifMIB ifAlias** should also not be modified.

Payload size, jitter buffer and filler patterns

This section talks about the options for setting payload size, jitter buffer size and filler patterns. However if you give the payload size without the jitter buffer size the jitter buffer size will be automatically determined.

Payload size and jitter buffer configuration

You can adjust the **payload** size and the **jitterbuffer** in the **pwe-tdm add** or **pwe-tdm modify** commands ([PWE commands on page 576](#)). You can adjust payload size and jitter buffer separately, however if you only use the payload size the jitterbuffer size will be configured based on that value automatically.

You can also use the **pwe-tdm calc** command ([pwe-tdm calc on page 586](#)) to calculate the optimal jittermean based on the payload size, or the optimal payload size based on the jittermean buffer size.

T1 payload size and jittermean calculation example.

```

zSH> pwe-tdm calc linetype t1 payload 250
jittermean = 2110 for payload = 250 (pct=1302, pdv=1460,
ats=24)

```

```

zSH> pwe-tdm calc linetype t1 jittermean 2100
payload = 247 for jittermean = 2100 (pct=1286, pdv=1458,
ats=24)

```

E1 payload size and jittermean calculation example. zSH>

```

pwe-tdm calc linetype e1 payload 200
jittermean = 1800 for payload = 200 (pct=781, pdv=1420,
ats=32)

```

```
zSH> pwe-tdm calc linetype e1 jittermean 1779
payload = 192 for jittermean = 1779 (pct=750, pdv=1414,
ats=32)
```

The **payload** parameter of the **pwe-tdm add** command is the size in bytes of the TDM (time division multiplexed) payload from the T1/E1 circuit inserted into PWE IP/UDP frames. The default **payload** value is 192 bytes. Acceptable payload range values are from 192 to 250 bytes. Both sides of the PWE service must be set to the same payload size.

The **jittermean** value is the mean/average jitterbuffer in microseconds from 0 to 170000. The default **jittermean** value for T1 with the default payload of 192 bytes is 1914 microseconds. The default **jittermean** value for E1 with the default payload of 192 bytes is 1779 microseconds.

If the **pwe-tdm add** command is used with a **payload** parameter and without the **jittermean**, the **jittermean** will automatically be set to an optimal value based on **pwe-tdm calc** results. It is recommended to have the system set jittermean automatically.

If no **payload** or **jittermean** values are set in the **pwe-tdm add** command, the **payload** defaults to 192 bytes and the **jittermean** defaults to 12500 microseconds (these default values are the same for both T1 and E1 mode).

Filler patterns for jitter buffer overflow/underflow

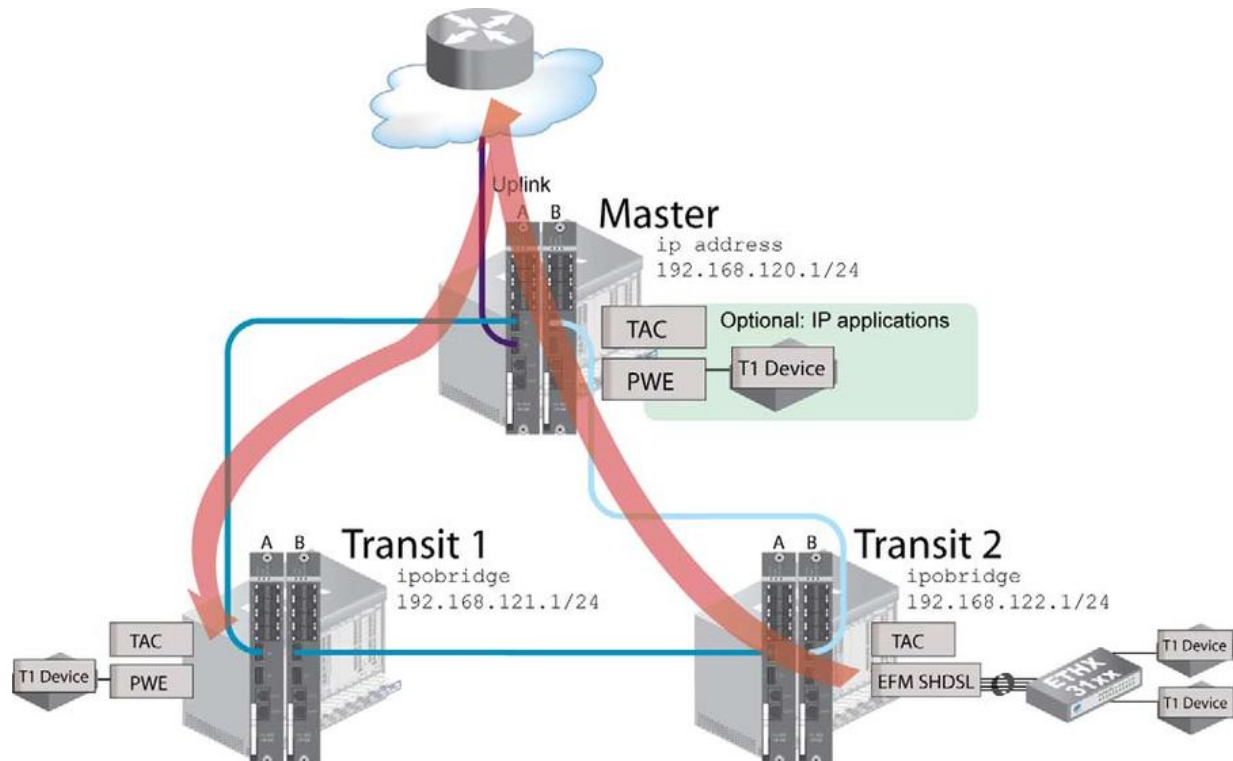
The value for the **replacepolicy** keyword defines what values to fill when there is a jitter buffer overflow or underflow.

- **allones**
Creates and implements a filler pattern of all ones.
- **filler**
Uses the filler pattern, defined in the value for the **pattern** keyword (values can be from 0-255)

PWE solution with EAPS

EAPS is a layer 2 bridging based solution and PWE solutions require a layer 3 IP address to define the far end of the PWE connection. To accomplish the combination of Layer 2, bridging and Layer 3, IP solutions, IP on a bridge is used. Rather than putting an IP address on an uplink as is shown in the configuration examples in this chapter, an *ipobridge* interface is added to the bridges on the EAPS nodes.

Figure 79: To combine PWE with EAPS use ipobridge interfaces on the uplinks



When a PWE device in a transit node in an EAPS ring, needs to access another PWE device on a transit node, the *ipobridge* interface address is given. The packet stream goes up to a router in the cloud, then back to the appropriate PWE device. For PWE devices outside of the EAPS ring, you address the target IP address as normal.

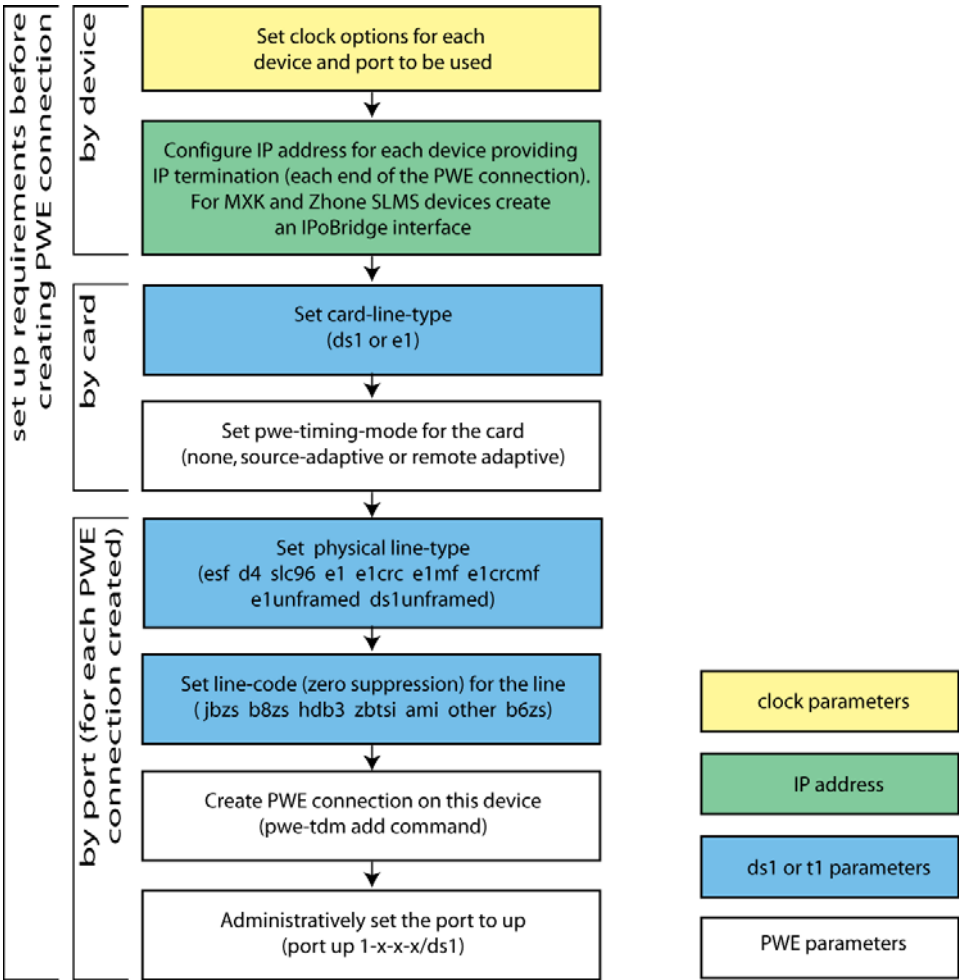


Note: PWE connections which require IP termination on the MXK (such as the PWE ports from T1/E1 PWE card) use IPv4 for IP addressing.

For information configuring EAPS with IP on a bridge, see [IP applications using IP on a bridge with EAPS](#).

Creating PWE connections

Figure 80: Overview of creating a PWE connection. This process (or similar process, depending on the devices involved) would be done on both ends of the PWE connection



PWE with T1 or E1

Using E1 or T1 does not change the PWE related commands. Both work the same in regards to configuration and the other topics discussed within this chapter.

Configuration of the other devices will match the E1 or T1 technology of the MXK in the scenario.

Configuring for T1 or E1 on the MXK

Both ends of the PWE connection must be in agreement regarding T1 and E1 and both settings below must also be in agreement.

- 1 Set the **card-line-type** parameter in the **card-profile** of the T1/E1 PWE card to either **ds1** (for T1) or **e1** (for E1)
- 2 Set the line-type parameter in the ds1 profile to **ds1unframed** (for T1) or **e1unframed** (for E1) or other acceptable line type parameter.

There are other acceptable **ds1** (or **e1**) **line-type** parameters other than the **ds1unframed** or **e1unframed**.

Table 45: ds1/e1 line-type parameters

line-type	description
other	
esf	Extended super frame (24 consecutive 193-bit frames)
d4	24 bytes of 8 bits each
slc96	
e1	
e1crc	
e1crcmt	
e1unframed	
ds1unframed	

PWE with CESoP channelization

This section provides an example of how to configure CESoP on the MXK PWE T1/E1 line card, and an ETHX-3100 end-point. In this example, the following information is assumed:

MXK IP Address: 192.168.3.1
 ETHX-3100 IP Address: 192.168.3.2
 CESoP Bundle: Timeslots 1-to-4

The EthX 31xx should be configured for E1 and adaptive mode



Note: The source IP of the MXK PWE bundles is the IP address given to the uplink to which you are connected.

In this example we will add a MXK PWE card. The IP address is already on the uplink.

- 1 Add the MXK PWE card and set the linetype for E1

```
zSH> card add 7 linetype e1
new card-profile 1/7/10215 added, sw-file-name
"mxlc24t1e1pwe.bin", 1 option: card-line-type e1
```

2 Configure the MXK PWE card for **source-adaptive** timing.

Notice that the card will need rebooting when the **pwe-timing-mode** is changed.

```
zSH> update card-profile 1/7/10215
card-profile 1/7/10215
Please provide the following: [q]uit.
sw-file-name: -----> {mxlc24t1e1pwe.bin}:
admin-status: -----> {operational}:
upgrade-sw-file-name: ---> {}:
upgrade-vers: -----> {}:
admin-status-enable: ----> {enable}:
sw-upgrade-admin: -----> {reloadcurrrev}:
sw-enable: -----> {true}:
sw-upgrade-enable: -----> {false}:
card-group-id: -----> {0}:
hold-active: -----> {false}:
weight: -----> {nopreference}:
card-line-type: -----> {e1}:
card-atm-configuration: -> {notapplicable}:
card-line-voltage: -----> {not-used}:
maxvpi-maxvci: -----> {notapplicable}:
card-init-string: -----> {}:
wetting-current: -----> {disabled}:
pwe-timing-mode: -----> {none}: source-adaptive
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Changing the pwe-timing-mode will result in a slot
reboot.
Continue? [y]es or [n]o: yes
pwe-timing-mode changed, slot 7 is rebooting ...
Record updated.
```

3 Update the **ds1-profile** for the connection, changing the **line-type** to **e1crc** and the **transmit-clock-source** to **looptiming**.

```
zSH> update ds1-profile 1/7/1/0/ds1
ds1-profile 1/7/1/0/ds1
Please provide the following: [q]uit.
line-type: -----> {e1}: e1crc
line-code: -----> {hdb3}:
send-code: -----> {sendnocode}:
circuit-id: -----> {e1}:
loopback-config: -----> {nolooop}:
signal-mode: -----> {none}:
fdl: -----> {fdlnone}:
dsx-line-length: -----> {dsx0}:
line-status_change-trap-enable: ---> {enabled}:
channelization: -----> {disabled}:
ds1-mode: -----> {other}:
csu-line-length: -----> {csu00}:
clock-source-eligible: -----> {eligible}:
```



```

transmit-clock-source: -----> {throughtiming}:
looptiming
cell-scramble: -----> {true}:
coset-polynomial: -----> {true}:
protocol-emulation: -----> {network}:
signal-type: -----> {loopstart}:
ds1-group-number: -----> {0}:
line-power: -----> {disabled}:
timeslot-assignment: ----->
{0+1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20+
21+22+23+24+25+26+27+28+29+30}:
transmit-clock-adaptive-quality: --> {stratum3}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

4 Configure the PWE bundle, including disabling ISDN

```

zSH> pwe-tdm add 1-7-1-0/ds1 srcip 192.168.3.1 srcudp
57001 destip 192.168.3.2 destudp 2142 tos 7 channels
1+2+3+4 payload 188 jittermean 5000 isdn disabled
Created 1-7-1-0-ds1-1/ds0bundle

```

Show the changes.

```

zSH> pwe-tdm show entry 1-7-1-0-ds1-1/ds0bundle
Pw Entry Config for PW 1-7-1-0-ds1-1/ds0bundle
type: -----> {basicCesPsn}
destip: -----> {192.168.3.2}
destudp: -----> {2142}
srcip: -----> {192.168.3.1}
srcudp: -----> {57001}
desc: -----> {}
channels: -----> {1+2+3+4}
tos: -----> {7}
adminstat: -----> {up}
payload: -----> {188}
jittermean: -----> {5000}
replacepolicy: -> {allones}
pattern: -----> {255}
isdn: -----> {disabled}

```



Note: The *channels* option in the **pwe-tdm** command above indicates that this will be configured in CESoP mode. If the channels option is not specified, the PWE will operate in SAToP mode by default.

5 Set the port to up.

```

zSH> port up 1-7-1-0-ds1-1/ds0bundle
1-7-1-0-ds1-1/ds0bundle set to admin state UP

zSH> port up 1-7-1-0/ds1

```

```
1-7-1-0/ds1 set to admin state UP
```

```
zSH> pwe-tdm modify entry 1-7-1-0-ds1-1/ds0bundle
adminstat up
```

- 6 Configure the ETHX-3100 for **remote-adaptive** timing, CESoP with timeslots **1-to-4**, and **disable** ISDN.

Configuring PWE for E1 ISDN PRI

This section provides an example of how to configure the MXK PWE line card and ETHX-3100 for E1 ISDN PRI operation. In this example, the following information is assumed:

MXK IP Address:	192.168.3.1
ETHX-3100 IP Address:	192.168.3.2
CESoP Bundle:	Timeslots 1-to-31 (required for E1 PRI support)

The EthX 31xx should be configured for E1 and adaptive mode.

This example, like the previous example, assumes there is no PWE card added in the system. The IP address is already set.

- 1 Add the MXK PWE card and set the linetype for E1

```
zSH> card add 7 linetype e1
new card-profile 1/7/10215 added, sw-file-name
"mxlc24t1elpwe.bin", 1 option: card-line-type e1
```

- 2 Configure the MXK PWE card for **source-adaptive** timing.

Notice that the card will need rebooting when the **pwe-timing-mode** is changed.

```
zSH> update card-profile 1/7/10215
card-profile 1/7/10215
Please provide the following: [q]uit.
sw-file-name: -----> {mxlc24t1elpwe.bin}:
admin-status: -----> {operational}:
upgrade-sw-file-name: ---> {}:
upgrade-vers: -----> {}:
admin-status-enable: ----> {enable}:
sw-upgrade-admin: -----> {reloadcurrrev}:
sw-enable: -----> {true}:
sw-upgrade-enable: -----> {false}:
card-group-id: -----> {0}:
hold-active: -----> {false}:
weight: -----> {nopreference}:
card-line-type: -----> {e1}:
card-atm-configuration: -> {notapplicable}:
card-line-voltage: -----> {not-used}:
maxvpi-maxvci: -----> {notapplicable}:
card-init-string: -----> {}:
```

```

wetting-current: -----> {disabled}:
pwe-timing-mode: -----> {none}: source-adaptive
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Changing the pwe-timing-mode will result in a slot
reboot.
Continue? [y]es or [n]o: yes
pwe-timing-mode changed, slot 7 is rebooting ...
Record updated.

```

3 Update the **ds1-profile** for the connection, changing the **line-type** to **e1crc** and the **transmit-clock-source** to **looptiming**.

```

zSH> update ds1-profile 1/7/1/0/ds1
ds1-profile 1/7/1/0/ds1
Please provide the following: [q]uit.
line-type: -----> {e1}: e1crc
line-code: -----> {hdb3}:
send-code: -----> {sendnocode}:
circuit-id: -----> {e1}:
loopback-config: -----> {nolooop}:
signal-mode: -----> {none}:
fdl: -----> {fdlnone}:
dsx-line-length: -----> {dsx0}:
line-status_change-trap-enable: ---> {enabled}:
channelization: -----> {disabled}:
ds1-mode: -----> {other}:
csu-line-length: -----> {csu00}:
clock-source-eligible: -----> {eligible}:
transmit-clock-source: -----> {throughtiming}:
looptiming
cell-scramble: -----> {true}:
coset-polynomial: -----> {true}:
protocol-emulation: -----> {network}:
signal-type: -----> {loopstart}:
ds1-group-number: -----> {0}:
line-power: -----> {disabled}:
timeslot-assignment: ----->
{0+1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20+
21+22+23+24+25+26+27+28+29+30}:
transmit-clock-adaptive-quality: --> {stratum3}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

4 Configure the PWE bundle for ISDN Line Termination (**lt**) using source-adaptive timing.

```

zSH> pwe-tdm add 1-7-1-0/ds1 srcip 192.168.3.1 srcudp
57001 destip 192.168.3.2 destudp 2142 tos 7 payload
188 jittermean 5000 isdn lt channels
1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20+21+
22+23+24+25+26+27+28+29+30+31
Created 1-7-1-0-ds1-1/ds0bundle

```

Show the changes.

```
zSH> pwe-tdm show entry 1-7-1-0-ds1-1/ds0bundle
Pw Entry Config for PW 1-7-1-0-ds1-1/ds0bundle
type: -----> {basicCesPsn}
destip: -----> {192.168.3.2}
destudp: -----> {2142}
srcip: -----> {192.168.3.1}
srcudp: -----> {57001}
desc: -----> {}
channels: --->
{1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20+21
+22+23+24+25+26+27+28+29+30+31}
tos: -----> {7}
adminstat: -----> {up}
payload: -----> {188}
jittermean: -----> {5000}
replacepolicy: -> {allones}
pattern: -----> {255}
isdn: -----> {1t}
```



Note: The *channels* option in the **pwe-tdm** command above indicates that this will be configured in CESoP mode. If the *channels* option is not specified, the PWE will operate in SAToP mode by default.

5 Set the port to up.

```
zSH> port up 1-7-1-0-ds1-1/ds0bundle
1-7-1-0-ds1-1/ds0bundle set to admin state UP

zSH> port up 1-7-1-0/ds1
1-7-1-0/ds1 set to admin state UP

zSH> pwe-tdm modify entry 1-7-1-0-ds1-1/ds0bundle
adminstat up
```

6 Configure the ETHX-3100 for ISDN Network Termination (**nt1**), **remote-adaptive** timing, and a CESoP bundle with timeslots **1-to-31**.

Admin up the PWE adminstat and port

The following commands administratively “up” the PWE connections and the ds1 ports.

```
zSH> pwe-tdm modify entry 1-1-1-0/ds1 adminstat up
zSH> pwe-tdm modify entry 1-1-2-0/ds1 adminstat up
zSH> pwe-tdm modify entry 1-1-3-0/ds1 adminstat up
...
zSH> pwe-tdm modify entry 1-1-84-0/ds1 adminstat up
zSH> port up 1-1-*-0/ds1
```

```
zSH> pwe-tdm add 1-1-1-0/ds1 srcip 10.100.100.218 destip  
10.100.100.118 srcudp 57101 destudp 57201 tos 224 payload  
192  
zSH> pwe-tdm add 1-1-2-0/ds1 srcip 10.100.100.218 destip  
10.100.100.118 srcudp 57102 destudp 57202 tos 224 payload  
192  
zSH> pwe-tdm add 1-1-3-0/ds1 srcip 10.100.100.218 destip  
10.100.100.118 srcudp 57103 destudp 57203 tos 224 payload  
192  
...  
zSH> pwe-tdm add 1-1-84-0/ds1 srcip 10.100.100.218 destip  
10.100.100.118 srcudp 57184 destudp 57284 tos 224 payload  
192
```

PWE alarms, logs and traps

This section discusses the following alarms:

- [PWE Loss of Service alarm, page 570](#)
- [PWE service degradation alarm, page 570](#)

PWE Loss of Service alarm

The PWE Loss of Service (LOS) alarm is cleared when the service is restored, the PWE port is set to **admin** down, or the PWE port is deleted.

If the **admin** status for the connection is set to disabled the alarm will not be set and the alarm will be cleared if already set.

PWE LOS logs

For the PWE LOS alarm there are three PWE log entries: PWE loss of service, PWE service restored, and PWE deleted.

PWE LOS traps

There are two traps; pwDown and pwUp.

Troubleshooting LOS

Check the operational and local or far-end statuses. Checking the status provides information about the location of the problem causing the condition. see [PWE operational status](#) for the **pwe-tdm stats status** command.

PWE service degradation alarm

A PseudoWire (PWE) alarm indication signal provides an alert when there is degradation of PWE services. The degraded service alarm is defined by parameters for the **pwe-tdm add/modify** command.

The degradation condition is the sum of the current missing packets, current buffer underruns, current packets misordered and dropped, and current malformed packets divided by the total incoming packets.

If the degradation condition exists for a configured window of time (when no degradation alarm is present), then a degradation alarm timer begins. If that timer reaches the amount of time set for the alarm threshold and the degradation condition is still present, then the PWE service degradation alarm is activated.

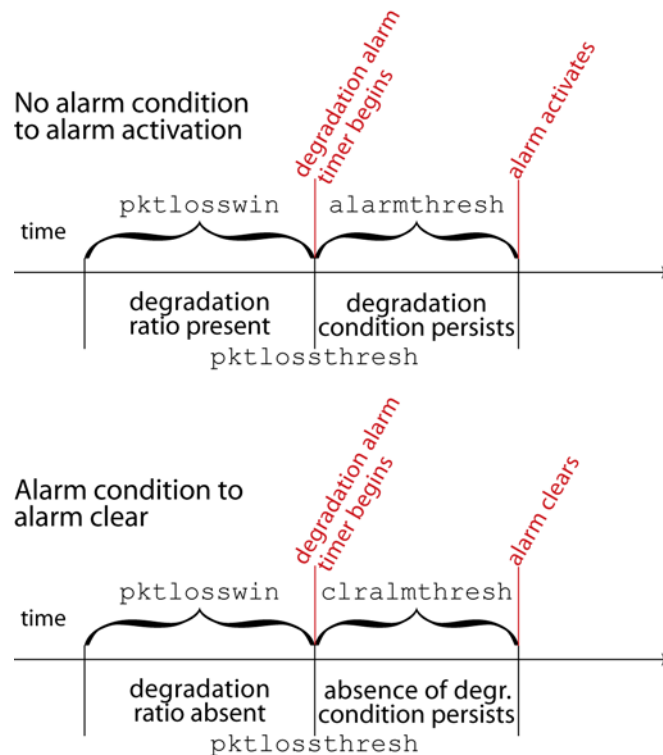
For clearing the alarm a similar process is used which also uses the initial configured window of time and the degradation condition. If the degradation condition is absent for the window of time (when the degradation alarm is active), then the degradation alarm timer begins. If that timer reaches the

amount of time set for the alarm clear and the degradation condition is still absent then the PWE service degradation alarm is cleared.

The configurable conditions are

- **pktlossthresh (%)**
a threshold of packets loss (a ratio of packets lost). Higher than this threshold value and the degradation condition is present. Below this threshold value and the degradation condition is absent.
- **pktlosswin (milliseconds)**
a window of time (in milliseconds) in which the threshold condition is present before alarm timing starts before (or the amount of time the threshold condition is absent before the timing for clearing the alarm).
- **alarmthresh (milliseconds)**
the amount of time (in milliseconds) when the degradation condition persists (beyond the trigger window) before the alarm is activated.
- **alarmthresh (milliseconds)**
the amount of time (in milliseconds) when the degradation condition is absent (beyond the trigger window) before the alarm is cleared.

Figure 81: PWE service degradation alarm parameters.





Note: PWE stats are polled in seconds. So while the thresholds and window are set in milliseconds, the actual timing will be rounded down to the nearest second unless it is less than one second in which case it will be rounded up to one second; 1000 ms = 1 second. If **pktlosswin**, **alarmthresh**, or **clralmthresh** are set to zero, they are rounded up to one second. If **pktlossthresh** is set to zero the alarm is disabled for that port.

PWE operational status

The **pwe-tdm stats status** CLI command displays operational and PWE status information.

```
zSH> pwe-tdm stats 1-5-1-0/ds1 status
```

PW Name	Create Time	Up Time	Last Change Time	Oper Stat	Local PW Status
					N S S P P
					F R T R T
1-5-1-0/ds1	563366094	298838680	563366095	up	D X X X X

Table 46: The pwe-tdm stats status display

Column	Status	Description
PW Name		The name of the PWE port
Create Tme		The time (from the MIB object sysUpTime) when the PWE connection was created.
Up Time		The amount of time that the PWE connection has been up.
Last Change Time		The time of the last change to the PWE connection.
Oper Stat		Shows the Operational Status of the local PWE port
	up	The port is up

Table 46: The pwe-tdm stats status display

Column	Status	Description
	down	The port is down. It is set to down when - It is detected that the port is not receiving packets from the far end PWE device pwOperStatus = DOWN and pwRemoteStatus = PSNPWTXFAULT and pwLocalStatus = PSNPWRXFAULT - An LOS or LOF or AIS condition is detected on the local TDM link pwOperStatus = DOWN and pwRemoteStatus = SERVICEPWRXFAULT and pwLocalStatus = SERVICEPWTXFAULT - The far-end PWE device indicates it is not receiving packets pwOperStatus = DOWN and pwRemoteStatus = PSNPWRXFAULT and pwLocalStatus = PSNPWTXFAULT - The far-end PWE device detects a LOS or LOF condition on the far-end TDM link, or an AIS condition on the far-end TDM link pwOperStatus = DOWN and pwRemoteStatus = SERVICEPWTXFAULT and pwLocalStatus = SERVICEPWRXFAULT
	testing	The admin status for the port is set to test mode
	lwrLayerDn	The lwrLayerDn state is when the only reason for not being in the 'up' state is either outer tunnel or physical layer down of the network side is in the down state
Local PW Status		Shows the operational status of the PWE port
	NFD	pwNotForwarding. The PWE port is not forwarding
	SRX	servicePwRxFault. A receive fault is detected on the PWE service

Table 46: The pwe-tdm stats status display

Column	Status	Description
	STX	servicePwTxFault. A transmit fault is detected on the PWE service
	PRX	psnPwRxFault. A PWE receive fault is detected
	PTX	psnPwTxFault. A PWE transmit fault is detected

PWE commands

This section includes the following commands:

- [*pwe-tdm add*](#)
- [*pwe-tdm delete*](#)
- [*pwe-tdm modify entry*](#)
- [*pwe-tdm modify txclock*](#)
- [*pwe-tdm show*](#)
- [*pwe-tdm stats*](#)
- [*pwe-tdm history*](#)
- [*pwe-tdm calc*](#)

pwe-tdm add

Add a PWE connection.

Syntax `pwe-tdm add <shelf-slot-port-subport/ds1> < srcip VALUE
srcudp VALUE destip VALUE destudp VALUE tos VALUE >
[channels VALUE desc VALUE | payload VALUE | jittermean
VALUE | replacepolicy VALUE | pattern VALUE | tos VALUE
| isdn VALUE]`

srcip

Source node IP address of a PWE connection.

srcudp

Source UDP port.

destip

Peer node IP address.

destudp

Destination UDP port

channels

The ds0 channels in the bundle

desc

A brief description of this pseudowire, may be a quoted string.

payload

Packet PayLoad Size (bytes). Any other size will be considered malformed.

jittermean

Mean jitter, or half the packet queueing buffer size in microseconds. The mean/average jitterbuffer in microseconds {0..170000}. Default value is 12500 microseconds.

replacepolicy

Value played out when CE bound packets over/underflow the jitter buffer, or are missing for any reason.

pattern

Filler byte pattern played out on the TDM interface if replacepolicy was set to filler.

tos

Pseudowire Type of Service.

isdn

Type of ISDN termination

pktlosswin

A window of time (in milliseconds) in which the threshold condition is present before alarm timing starts before (or the amount of time the threshold condition is absent before the timing for clearing the alarm).

pktlossthresh

A threshold of packets loss (a ratio of packets lost). Higher than this threshold value and the degradation condition is present. Below this threshold value and the degradation condition is absent.e alarm).

alarmthresh

The amount of time (in milliseconds) when the degradation condition persists (beyond the trigger window) before the alarm is activated.

clralmthresh

the amount of time (in milliseconds) when the degradation condition is absent (beyond the trigger window) before the alarm is cleared.

Example Example: UDP mapped pairs with default payload and jittermean
local PWE source IP/UDP = remote PWE destination IP/UDP, and
local PWE destination IP/UDP = remote PWE source IP/UDP

MXK-1MXK-2

LOCAL PWE [IP 10.1.1.1] REMOTE PWE [IP 10.1.1.8]

SRC IP 10.1.1.1SRC IP 10.1.1.8

SRC UDP 59101SRCUDP 58222

DEST IP 10.1.1.8DEST IP 10.1.1.1

DEST UDP 58222 DEST UDP 59101

MXK 1 example

```
zSH> pwe-tdm add 1-3-2-0/ds1 srcip 10.1.1.1 srcudp 59101
destip 10.1.1.8 destudp 58222 tos 224
```

MXK 2 example

```
zSH> pwe-tdm add 1-6-11-0/ds1 srcip 10.1.1.8 srcudp
58222 destip 10.1.1.1 destudp 59101 tos 224
```

pwe-tdm delete

Delete a PWE interface.

Syntax `pwe-tdm delete <shelf-slot-port-subport/name>`

For CESoP entry:

```
pwe-tdm delete <shelf-slot-port-subport-ds1-bundle/
ds0bundle>
```

Example Delete PWE entry on slot 3 port 5

```
zSH> pwe-tdm delete 1-3-5-0/ds1
```

pwe-tdm modify entry

Modify an existing PWE object.

Syntax **Syntax**

```
pwe-tdm modify entry <shelf-slot-port-subport/ds1>
< srcip VALUE | srcudp VALUE | destip VALUE | destudp
VALUE | channels VALUE | tos VALUE | adminstat VALUE |
desc VALUE | payload VALUE | jittermean VALUE |
replacepolicy VALUE | pattern VALUE | tos VALUE | isdn
VALUE >
```

For CESoP entry:

Syntax

```
pwe-tdm modify entry <shelf-slot-port-subport-ds1-bundle/
ds0bundle> < srcip VALUE | srcudp VALUE | destip VALUE |
destudp VALUE | channels VALUE | tos VALUE | adminstat
VALUE | desc VALUE | payload VALUE | jittermean VALUE |
replacepolicy VALUE | pattern VALUE >
```

srcip

Source node IP address of a PWE connection.

srcudp

Source UDP port.

destip

Peer node IP address.

destudp

Destination UDP port

channels

The ds0 channels in the bundle

desc

A brief description of this pseudowire, may be a quoted string.

payload

Packet PayLoad Size (bytes). Any other size will be considered malformed.

jittermean

Mean jitter, or half the packet queueing buffer size in microseconds. Approximately equal to PDV.

replacepolicy

Value played out when CE bound packets over/underflow the jitter buffer, or are missing for any reason.

pattern

Filler byte pattern played out on the TDM interface if replacepolicy was set to filler.

tos

Pseudowire Type of Service.

isdn

Type of ISDN termination

pktlosswin

A window of time (in milliseconds) in which the threshold condition is present before alarm timing starts before (or the amount of time the threshold condition is absent before the timing for clearing the alarm).

pktlossthresh

A threshold of packets loss (a ratio of packets lost). Higher than this threshold value and the degradation condition is present. Below this threshold value and the degradation condition is absent.e alarm).

alarmthresh

The amount of time (in milliseconds) when the degradation condition persists (beyond the trigger window) before the alarm is activated.

clralmthresh

the amount of time (in milliseconds) when the degradation condition is absent (beyond the trigger window) before the alarm is cleared.

Example Change the payload to 250 and jittermean to 2111 [as calculated with pwe-tdm calc]

```
zSH> pwe-tdm calc payload 250
jittermean = 2111 for payload = 250 (pdv=1460, pct=1302,
ats=24)

zSH> pwe-tdm modify entry 1-3-5-0/ds1 payload 250
jittermean 2111
```

pwe-tdm modify txclock

Modify transmit clock information configuration parameters required to transmit and recover pwe packet timing.

Syntax `pwe-tdm modify txclock < PORTSPEC > < [[sourcetiming]
< loop | local | through >] | [[quality] < stratum1 |
stratum3 | stratum3e | stratum4 >] >`

sourcetiming

Describes where this port's txclock is derived.

loop

The txclock derived from the port's rxclock.

local

The txclock is derived from on board local oscillator.

through

The txclock derived from clock provided to chassis either from an external clock, either from other t1/e1 ports or a bits clock.

quality

Determines sync drift when operating in pwe adaptive mode. The options are **stratum1**, **stratum3**, **stratum3e** or **stratum4** as based on ANSI Standard T1.101 reference for clock quality.

pwe-tdm show

The pwe-tdm show command displays profile information for the pseudowire object.

Syntax `pwe-tdm show entry [NAMESPEC] [srcip | srcudp | destip
| destudp | channels | adminstat | type | desc | payload |
jittermean | replacepolicy | pattern | tos | isdn]`

For CESoP entry:

`pwe-tdm show entry <shelf-slot-port-subport-ds1-bundle/
ds0bundle> [srcip | srcudp | destip | destudp | channels
| adminstat | type | desc | payload | jittermean |
replacepolicy | pattern | tos | isdn]`

srcip

Source node IP address of a PWE connection.

srcudp

Source UDP port.

destip

Peer node IP address.

destudp

Destination UDP port

channels

The ds0 channels in the bundle

adminstat

The administrative status of this pseudowire.

type

The emulated service carried over this pseudowire.

desc

A brief description of this pseudowire, may be a quoted string.

payload

Defines the Packet PayLoad Size in bytes. Any other size will be considered malformed.

jittermean

Mean jitter, or half the packet queueing buffer size in microseconds. Approximately equal to PDV.

replacepolicy

Value played out when CE bound packets over/underflow the jitter buffer, or are missing for any reason.

pattern

Filler byte pattern played out on the TDM interface if replacepolicy was set to filler.

tos

Pseudowire Type of Service.

isdn

Type of ISDN termination

```

Example zSH> pwe-tdm show entry 1-3-5-0/ds1Pw Entry Config for PW
1-3-5-0/ds1
type: -----> {t1Satop}
destip: -----> {192.16.78.105}
destudp: -----> {2142}
srcip: -----> {192.16.78.200}
srcudp: -----> {57641}
desc: -----> {}
channels: -----> {none}
tos: -----> {224}
adminstat: -----> {up}
payload: -----> {250}
jittermean: -----> {2111}
replacepolicy: -> {allones}
pattern: -----> {255}
isdn: -----> {disabled}

```

pwe-tdm stats

The pwe-tdm stats command provides configuration and current statistics for the pseudowire.

```

Syntax pwe-tdm stats [ NAMESPEC ] [ [ status | traffic | tdmpkts
| tdmsecs ] | [ createtime | uptime | lastchange |
operstat | pwstatus | inpackets | inbytes | outpackets |
outbytes | missingpkts | reordered | underruns | dropped
| malformed | errsecs | severrsecs | uasecs ] ]

```

status

The **status** keyword shows the following statistics in a table:

- createtime
- uptime
- lastchange
- operstat
- pwstatus

traffic

The **traffic** keyword shows the following statistics in a table:

- inpackets
- inbytes
- outpackets
- outbytes

tdmpkts

The **tdmpkts** keyword shows the following statistics and history in a table:

- missingpkts
- reordered
- underruns
- dropped
- malformed

tdmsecs

The **tdmsecs** keyword shows the following statistics and history in a table:

- errsecs
- severrsecs
- uasecs

createtime

SysUpTime when the pseudowire was created.

uptime

Time since the last change of operStatus to up.

lastchange

Time since the pseudowire entered its current operational state.

operstat

The operational status of the pseudowire

Local PW Status

The local status of the pseudowire

- NFD = pwNotForwarding
- SRX = servicePwRxFault (Sending frame with L indication)
- STX = servicePwTxFault (Receiving AIS on local ds1)
- PRX = psnPwRxFault (Sending frame with R indication)

- PTX = psnPwRxFault (Receiving frame with R indication)

pwstatus

The status of the pseudowire and the interfaces affecting the pseudowire.

inpackets

Packets received in the current 15-minute interval.

inbytes

Bytes received in the current 15-minute interval.

outpackets

Packets forwarded in the current 15-minute interval.

outhbytes

Bytes forwarded in the current 15-minute interval.

missingpkts

Missing packets as detected via control word sequence number gaps.

reordered

Packets detected out of sequence but successfully re-ordered.

underruns

Number of times a packet needed to be played out and the jitter buffer was empty.

dropped

Number of packets detected out of order which could not be re-ordered, or could not fit in the jitter buffer.

malformed

Number of packets detected with unexpected size, or bad header stack.

errsecs

Number of Errored Seconds encountered.

severrsecs

Number of Severely Errored Seconds encountered.

uasecs

Number of Unavailable Seconds encountered.

Example

```
zSH> pwe-tdm stats status
-----+-----+-----+-----
---+-----+-----
| | | |Local PW
| | | Last |Status
PW Name|Create| Up|Change|Oper|N S S P P
| Time|Time|Time|Stat|F R T R T
| | | |D X X X X
-----+-----+-----+-----
---+-----+-----+-----
1-6-1-0/ds1|15607|32721571|15609|down|X
1-6-9-0/ds1|24382557|8338941|24382652|down|X
```

```
zSH> pwe-tdm stats traffic
```

```
-----+-----+-----+-----
-----+-----
|          | In      | Out   | In
| Out      |          |       |
| PW Name  | Packets | Packets | Bytes
| Bytes    |         |         |
-----+-----+-----+-----
-----+-----
| 1-6-1-0/ds1 | 3441462 | 252552922 |
1032438600 | 2751432568
| 1-6-9-0/ds1 | 0 | 83905734 |
0 | 3125318444
```

```
zSH> pwe-tdm stats tdmPkts
```

```
-----+-----+-----+-----
--++-----+-----
|          | Missing | Reorder |
Underrun | Dropped | Malform
-----+-----+-----+-----
--++-----+-----
| 1-6-1-0/ds1 | 0 | 0 | 6 | 120 | 0
| 1-6-9-0/ds1 | 0 | 0 | 0 | 0 | 0
```

```
zSH> pwe-tdm stats tdmsecs
```

```
-----+-----+-----+-----
-----
|          | Errorred | Severe |
Unavail
-----+-----+-----+-----
-----
| 1-6-1-0/ds1 | 48141 | 0 |
0
| 1-6-9-0/ds1 | 7856 | 0 |
0
```

```
zSH> pwe-tdm stats
```

```
Pw Entry Stats for PW 1-6-1-0/ds1
```

```
createtime: --> {15607}
uptime: -----> {32712362}
lastchange: --> {15609}
operstat: ----> {down}
pwstatus: ----> {psnPwRxFault}
inpackets: ---> {3305048}
outpackets: --> {252416507}
inbytes: -----> {991514400}
outbytes: ----> {2710508068}
missingpkts: -> {0}
reordered: ---> {0}
underruns: ---> {6}
dropped: -----> {120}
malformed: ---> {0}
errsecs: -----> {48141}
seversecs: --> {0}
```

```
uasecs: -----> {0}
```

The "pwe-tdm stats" output can be filtered by using the pwe-tdm stats command while specifying the correct identifier:

```
zSH> pwe-tdm stats 1-6-1-0/ds1 createtime
```

```
Pw Entry Stats for PW 1-6-1-0/ds1
```

```
createtime: --> {15607}
```

createtime is uplink uptime in hundredths of a second at time of pwe entry creation

uptime is uplink uptime in hundredths of a second at time of pwe entry last went ACT.

lastchange is uplink uptime in hundredths of a second at time of pwe entry last link change

pwe-tdm history

History statistics provide tracking of pwe-tdm traffic performance data over a 24-hour period. Data snapshots are taken at 15-minute intervals.

Syntax `pwe-tdm history < NAMESPEC | interval VALUE > [ [ packets | bytes | tdmpkts | tdmsecs ] | [ inpackets | inbytes | outpackets | outbytes | missingpkts | reordered | underruns | dropped | malformed | errsecs | severrsecs | uasecs ]`

interval

{1..96} The 1-based index of performance history data.

packets

Use this keyword to show tabular PW history of inpackets/outpackets

bytes

Use this keyword to show tabular PW history of inbytes/outbytes

Example The pwe-tdm history command can be used to display the last 24 hours of 15-minute statistics intervals per pwe port [up to 96 intervals]. Like the pwe-tdm stats command, the output can be filtered to a specified entry

```
zSH> pwe-tdm history 1-3-1-0/ds1 interval 2
Pw History Stats at PW 1-3-1-0/ds1 Interval 2
inpackets: ---> {694826}
outpackets: --> {694827}
inbytes: -----> {208447800}
outbytes: -----> {208448100}
missingpkts: -> {1}
reordered: ---> {0}
underruns: ---> {0}
dropped: -----> {0}
malformed: ---> {0}
```

```
errsecs: -----> { 0 }
severrsecs: --> { 0 }
uasecs: -----> { 0 }
```

pwe-tdm calc

The **pwe-tdm calc** macro finds the recommended mean jitter size given the packet payload, or the packet payload given the mean jitter size, for both SAToP and fractional PWE. If known, the packet delay value (PDV) can be input for a better estimate.

If the **pwe-tdm add** command is used with a **payload** parameter and without the **jittermean**, the **jittermean** will automatically be set to an optimal value based on **pwe-tdm calc** results. It is recommended to have the system set jittermean automatically.

If no **payload** or **jittermean** values are set in the **pwe-tdm add** command, the **payload** defaults to 192 bytes and the **jittermean** defaults to 12500 microseconds (these default values are the same for both T1 and E1 mode).

Syntax **pwe-tdm calc** < **linetype** **VALUE** > < **payload** **VALUE** |
jittermean **VALUE** > [**pdv** **VALUE**] [**numchannels** **VALUE**]

linetype

The linetype { t1 | e1 } is required.

payload

Packet PayLoad Size (bytes). Any other size will be considered malformed. The **payload** parameter of the **pwe-tdm add** command is the size in bytes of the TDM (time division multiplexed) payload from the T1/E1 circuit inserted into PWE IP/UDP frames.

The default **payload** value is 192 bytes. Acceptable payload range values are from 192 to 250 bytes. Both sides of the PWE service must be set to the same payload size.

jittermean

Mean jitter, or half the packet queueing buffer size in microseconds. Approximately equal to PDV. The **jittermean** value is the mean/average jitterbuffer in microseconds from 0 to 170000.

The default **jittermean** value for T1 with the default payload of 192 bytes is 1914 microseconds. The default **jittermean** value for E1 with the default payload of 192 bytes is 1779 microseconds.

pdv

Packet Delay Variation (microseconds).

numchannels

ds0 channels in the PWE bundle per ds1 port, stated as C/N where C is the number of channels and N = 24 (T1) or 32 (E1). Defaults to all channels.

The numchannels keyword is required for CESoP and must be from 1 to the maximum defined by the linetype. If the **numchannels** keyword is NOT supplied, SAToP is implied.

Example

```
zSH> pwe-tdm calc payload 250
jittermean = 2111 for payload = 250 (pdv=1460, pct=1302,
ats=24)
```

```
zSH> pwe-tdm calc jittermean 2000
payload = 217 for jittermean = 2000
```


8

LINK AGGREGATION CONFIGURATION

This chapter describes MXK link aggregation on Ethernet cards and includes:

- [Link aggregation overview, page 589](#)
- [Configure link aggregation on Ethernet uplink ports, page 594](#)
- [Configure link aggregation on Ethernet line card ports, page 601](#)
- [Configure link aggregation bridges, page 602](#)

Link aggregation overview

This section describes:

- [Link aggregation and LACP, page 590](#)
- [LACP link aggregation active mode, page 591](#)
- [Link resiliency, page 591](#)
- [Ethernet uplink ports available for link aggregation, page 591](#)

The MXK supports 802.3ad link aggregation on Ethernet uplink cards and Ethernet line cards.

Link aggregation allows aggregating physical 10 GE or 100/1000 Ethernet uplink ports into one single aggregated logical port for additional bandwidth, capacity, and resiliency. A link aggregation group can consist of up to eight ports.

Each link aggregation group is manually created by adding links to the group with the **linkagg add group** with **mode active** command:

```
linkagg <add|delete> group <aggregation name/type> link  
<linkname/type> mode <active|passive>|
```

For example:

```
zSH> linkagg add group 1-a-1-0/linkagg link 1-a-2-0/eth mode active  
Interface 1-a-1-0/linkagg does not exist  
Link aggregation successfully created.
```



Note: The 10 GE and the 100/1000 Ethernet ports cannot be aggregated into the same link aggregation group.



Note: Link aggregation across cards is only supported on redundant uplink cards that support CLK/TOP.

The MXK Ethernet cards that support link aggregation are:

- MXK-UPLINK-2X10G-8X1GE
- MXK-UPLINK-4X1GE-CU
- MXK-UPLINK-8X1GE
- MXK-UPLINK-4X1GE
- MXK-AE-2X10G-8X1GE
- MXK-AE20-FE/GE-2S
- MXK-AE20-FE/GE

The MXK Ethernet uplink cards that support link aggregation and link aggregation across cards are:

- MXK-UPLINK-2X10G-8X1G-TOP
- MXK-UPLINK-2X10G-8X1G-CLK
- MXK-UPLINK-6X1G-CLK

Link aggregation and LACP

The MXK uplink cards support Link Aggregation Control Protocol (LACP), a Layer 2 protocol used between network elements to exchange information regarding a link's ability to be aggregated with other similar links.

For redundant configurations, the Ethernet ports on both the active uplink card and the redundant uplink must be configured in *active* mode. This allows the link aggregated ports on each card to provide redundant uplink port protection.

For link aggregation across cards on the supported uplink cards, configuring the port in slot *a* for in *active* mode automatically configures the port in slot *b*.



Note: You must configure the Ethernet switch on the remote end for link aggregation.

lACP command

Use the **lACP** command to verify that the *aggregation partner key* number of the LACP enabled link aggregation group match, and view other link aggregation information.

lACP command syntax usage:

```
zSH> lACP
Usage: lACP <agg|id|monitor|state> [portNo] | lACP stats [portNo] [clear]
```

After connecting the MXK to an LACP enabled switch, you can verify that the *aggregation partner key* number matches for each link to the switch.

```
zSH> lacp monitor 2
PORT 2:
selected = SELECTED Disabled Traffic Enabled
actor state:7c
partner state:5a
1: partner key 4002, par port pri 8000, partner port # 2, actor state AGGREGATION
SYNCHRONIZATION COLLECTING DISTRIBUTING DEFAULTED, partner state LACP_TIMEOUT
SYNCHRONIZATION COLLECTING DEFAULTED
partner system: 00:00:00:00:00:00
1: agg id f03f5e0, par sys pri: 8000, agg partner key 4002
par sys: 00:00:00:00:00:00

zSH> lacp monitor 3
PORT 3:
selected = SELECTED Disabled Traffic Enabled
actor state:7c
partner state:5a
2: partner key 4002, par port pri 8000, partner port # 3, actor state
AGGREGATION SYNCHRONIZATION COLLECTING DISTRIBUTING DEFAULTED , partner state
LACP _TIMEOUT SYNCHRONIZATION COLLECTING DEFAULTED
partner system: 00:00:00:00:00:00
2: agg id f03f500, par sys pri: 8000, agg partner key 4002
par sys: 00:00:00:00:00:00
```

LACP link aggregation active mode

The *active* mode supports LACP and enables the Ethernet link to send and receive LACP messages.

Link resiliency

The link aggregation stays up as long as one link in the group is operational. Link aggregation manages links as they fail and come up again with no interruption in service. However, if all the links in a link aggregation group fail, the link aggregation group changes to a down state until a physical link is restored.

MXK Ethernet ports available for link aggregation

Ethernet uplink ports available for link aggregation

[Figure 82](#) and [Figure 83](#) illustrate the physical ports available on MXK uplink cards for link aggregation.

Figure 82: Ethernet uplink ports available for link aggregation

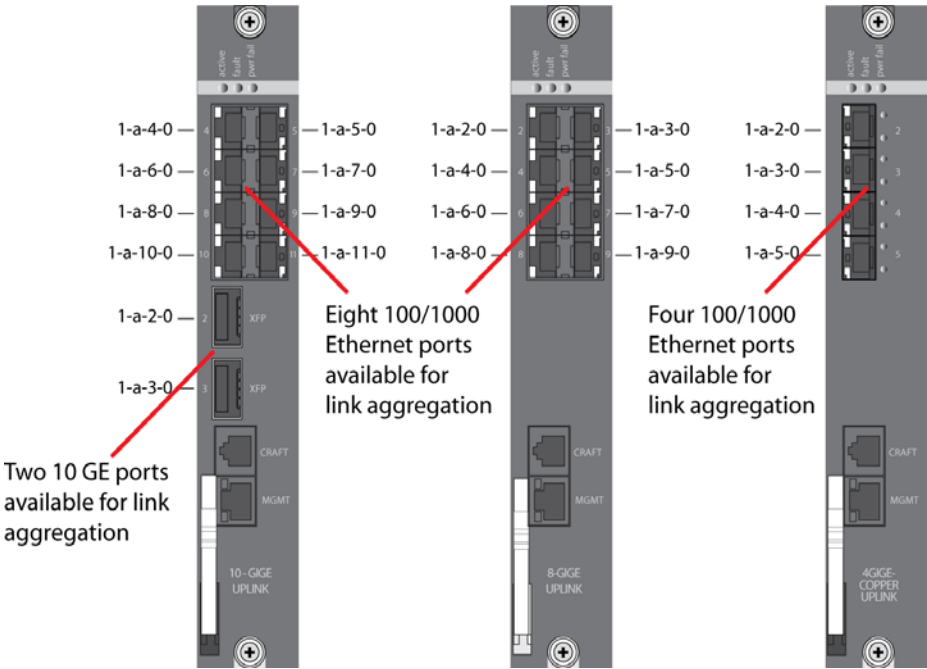
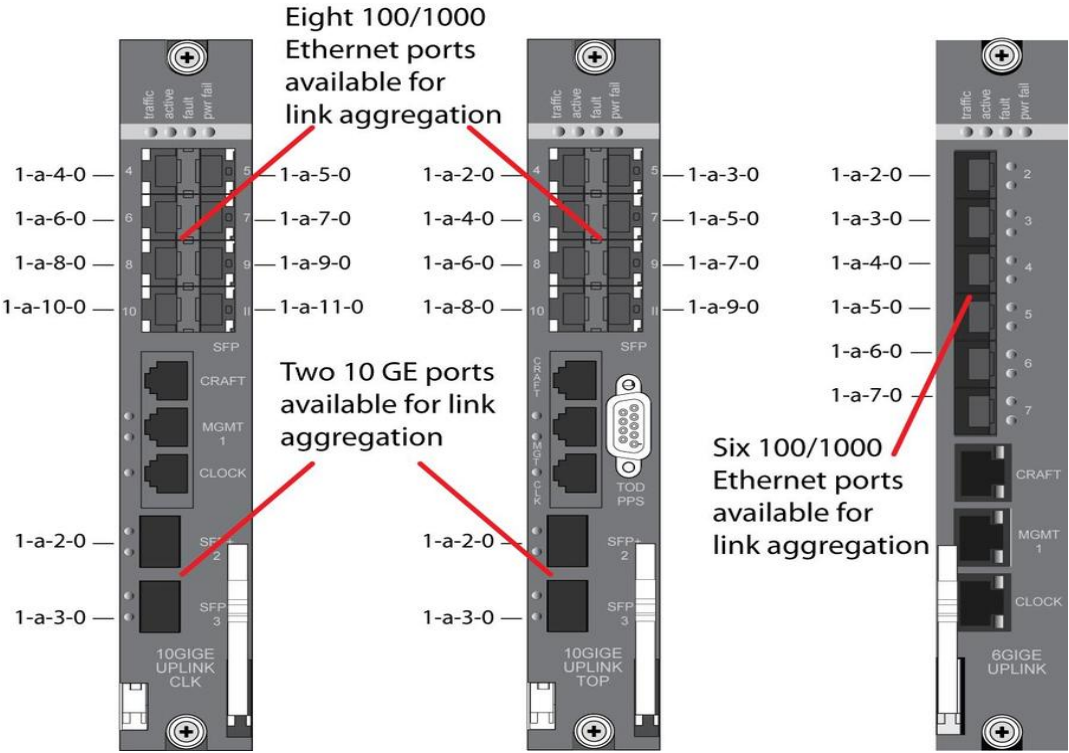


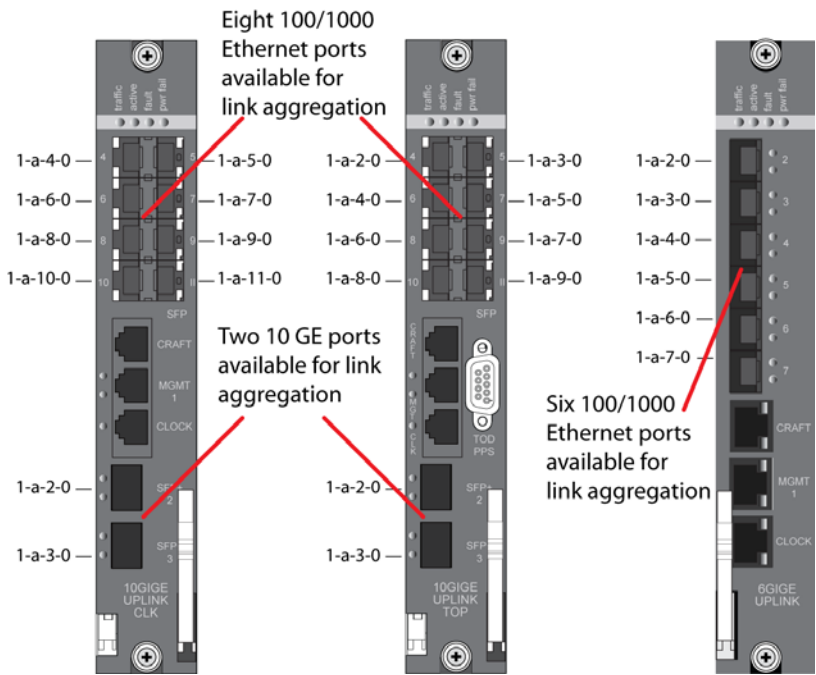
Figure 83: Ethernet uplink ports available for link aggregation on uplink cards with clock/TOP



Ethernet uplink card ports available for link aggregation across cards

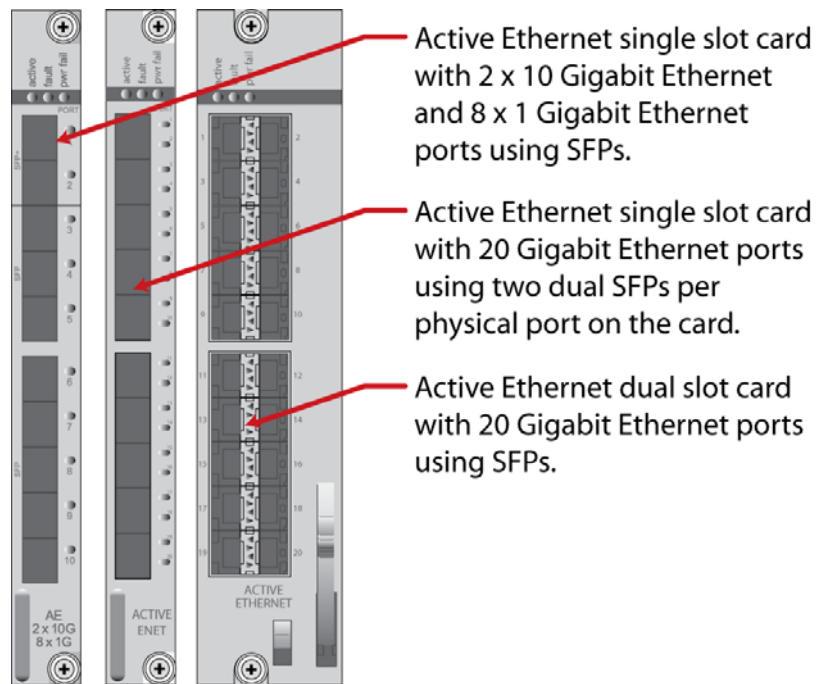
Figure 84 illustrates the uplink ports available when the **options** parameter in the **system 0** profile is set to *enablexcardlinkagg* to enable link aggregation across cards.

Figure 84: Ethernet uplink ports available for link aggregation across cards on uplink cards with clock/TOP



Ethernet line card ports available for link aggregation

Figure 85 illustrates the physical ports available on the MXK Ethernet line cards for link aggregation.

Figure 85: Line card Ethernet ports available for link aggregation

Configure link aggregation on Ethernet uplink ports

This section discusses:

- [Configure link aggregation on Ethernet uplink ports, page 594](#)
- [Configure multiple Ethernet uplink ports for link aggregation across cards, page 596](#)
- [Delete a link aggregation group, page 600](#)

Active mode in link aggregation means the link aggregation group is automatically created.

Configure a Ethernet uplink port for redundant link aggregation

When the aggregation mode on the Ethernet uplink port is *active*, the device sends and receives LACP and the redundant link aggregation is dynamic, i.e., the redundant link aggregation group is automatically created on both ports *a* and *b*.

Configuring a Ethernet uplink port for redundant link aggregation with LACP

Enable a Ethernet uplink port for redundant link aggregation with LACP.

- 1 Connect the MXK to the LACP enabled switch.

2 Create the redundant link aggregation by adding the link in *active* mode.

Linkagg groups are static and all links in the link aggregation group must be added with the **linkagg add group** command.

```
zSH> linkagg add group 1-a-1-0/linkagg link 1-a-2-0/eth mode active
Interface 1-a-1-0/linkagg does not exist
Link aggregation successfully created.
```

3 View the aggregation mode status.

```
zSH> get ether 1-a-2-0/eth
ether 1-a-2-0/eth
autonegstatus: ----> {enabled}
mauType: -----> {mau1000baselxfd}
restart: -----> {norestart}
ifType: -----> {mau1000baselxfd}
autonegcap: -----> {b10baseTFD+b100baseTXFD+b1000baseTFD}
remotefault: -----> {noerror}
clksrc: -----> {automatic}
pauseFlowControl: -> {disabled}
aggregationMode: --> {active}
linkStateMirror: --> {0/0/0/0/0}
maxFrameSize: -----> {0}
ingressRate: -----> {0}
ingressBurstSize: -> {0}
egressRate: -----> {0}
egressBurstSize: --> {0}
```

4 View the link aggregation.

```
zSH> linkagg show
LinkAggregations:
slot unit ifName          partner: Sys          Pri    grp ID  status  agg mode
-----
a*  1    1-a-1-0          00:00:00:00:00:00    0x0    0x0    ACT    Active
      links          slot  port  subport          status
-----
      1-a-2-0          a    2    0          ACT
b    1    1-b-1-0          00:00:00:00:00:00    0x0    0x0    DSA    Active
      links          slot  port  subport          status
-----
      1-b-2-0          b    2    0          DSA
global linkagg group red type: red
```

Configuring multiple Ethernet uplink ports for link aggregation with LACP

Enable two or more Ethernet ports on the uplink card for LACP.

- 1** Connect the MXK to the LACP enabled switch.
- 2** Add the Ethernet ports to the linkagg group and configure the mode of the ports to *active*.

Linkagg groups are static and all links in the link aggregation group must be added with the **linkagg add group** command.

```
zSH> linkagg add group 1-a-1-0/linkagg link 1-a-2-0/eth mode active
Interface 1-a-1-0/linkagg does not exist
Link aggregation successfully created.

zSH> linkagg add group 1-a-1-0/linkagg link 1-a-3-0/eth mode active
Link aggregation successfully created.
```

3 Verify the change to active.

```
zSH> get ether 1-a-3-0/eth
ether 1-a-3-0/eth
autonegstatus: ----> {enabled}
mauType: -----> {mau1000baselxfd}
restart: -----> {norestart}
ifType: -----> {mau1000baselxfd}
autonegcap: -----> {b10baseTFD+b100baseTXFD+b1000baseTFD}
remotefault: -----> {noerror}
clksrc: -----> {automatic}
pauseFlowControl: -> {disabled}
aggregationMode: --> {active}
linkStateMirror: --> {0/0/0/0/0}
maxFrameSize: -----> {0}
ingressRate: -----> {0}
ingressBurstSize: -> {0}
egressRate: -----> {0}
egressBurstSize: --> {0}
```

4 Verify the link aggregation group.

```
zSH> linkagg show
LinkAggregations:
slot unit ifName          partner: Sys          Pri    grp ID  status  agg mode
-----
a*  1  1-a-1-0      f8:66:f2:0d:3c:41    0xa    0xe    ACT    Active
    links      slot  port  subport            status
    -----
    1-a-2-0     a    2    0                ACT
    1-a-3-0     a    3    0                ACT
b   1  1-b-1-0      f8:66:f2:0d:3c:41    0xa    0xe    DSA    Active
    links      slot  port  subport            status
    -----
    1-b-2-0     b    2    0                DSA
    1-b-3-0     b    3    0                DSA
global linkagg group red type: red
```

Configure multiple Ethernet uplink ports for link aggregation across cards

Link aggregation across uplink cards can be configured on redundant uplink cards. After setting the link aggregation mode on the Ethernet uplink ports in

slot *a* to *active*, the MXK sends and receives LACP and link aggregation is dynamic, i.e. a link aggregation group is automatically created. All of the ports configured as *active* will be included in the link aggregation group.

When the Ethernet port in slot *a* is configured to *active*, the Ethernet port in slot *b* is automatically added to the link aggregation group.

Configuring Ethernet uplink ports for link aggregation across cards with LACP

Enable two or more Ethernet uplink ports in a redundant configuration for link aggregation across uplinks.

- 1 Verify that the MXK has redundant uplink cards.

```
zSH> slots
MXK 819
Uplinks
  a:*MXK SIX GIGE (RUNNING+TRAFFIC)
  b: MXK SIX GIGE (NOT_PROV)
Cards
  1: MXK GSHDSL-24 Bonded/with NTP (NOT_PROV)
  3: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (NOT_PROV)
  5: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (NOT_PROV)
 10: MXK 20 ACT ETH SS CSFP (NOT_PROV)
 11: MXK 20 ACT ETH (NOT_PROV)
 13: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (NOT_PROV)
```

- 2 Provision the uplink card in slot *b* if needed.

```
zSH> card add b group 1 linetype ds1
new card-profile 1/b/10130 added, sw-file-name "mxup6g.bin", 2 options:
card-group-id 1 card-line-type ds1
```

Verify the redundant cards are running.

```
zSH> slots
MXK 819
Uplinks
  a:*MXK SIX GIGE (RUNNING+TRAFFIC)
  b: MXK SIX GIGE (RUNNING)
Cards
  1: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
  3: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (RUNNING)
  5: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (RUNNING)
 10: MXK 20 ACT ETH SS CSFP (RUNNING)
 11: MXK 20 ACT ETH (RUNNING)
 13: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (RUNNING)
```

- 3 Update the **system 0** profile to enable link aggregation across uplinks by changing the **options** parameter to enablexcardlinkagg.

```
zSH> update system 0
system 0
Please provide the following: [q]uit.
```

```

syscontact: -----> {}:
sysname: -----> {}:
syslocation: -----> {}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {false}:
zmsconnectionstatus: --> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: --> {false}:
configsyncpriority: --> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: --> {}:
configsyncstatus: -----> {syncinitializing}:
configsyncuser: -----> {}:
configsyncpasswd: -----> {** private **}: ** read-only **
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {0.0.0.0}:
alternateipaddress: --> {0.0.0.0}:
countryregion: -----> {us}:
primaryclocksource: --> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}:
revertiveclocksource: -> {true}:
voicebandwidthcheck: --> {false}:
alarm-levels-enabled: -> {critical+major+minor+warning}:
userauthmode: -----> {local}:
radiusauthindex: -----> {0}:
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {NONE(0)}: enablexcardlinkagg
reservedVlanIdStart: --> {0}:
reservedVlanIdCount: --> {0}:
snmpVersion: -----> {snmpv2}:
persistentLogging: -----> {disabled}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Please reboot the system for enablexcardlinkagg change to take effect.
Record updated.

```

4 Verify the change to the **options** parameter.

```

zSH> get system 0
system 0
syscontact: -----> {}
sysname: -----> {}
syslocation: -----> {}
enableauthtraps: -----> {disabled}
setserialno: -----> {0}
zmsexists: -----> {false}
zmsconnectionstatus: --> {inactive}
zmsipaddress: -----> {0.0.0.0}
configsyncexists: -----> {false}

```

```

configsyncoverflow: ---> {false}
configsyncpriority: ---> {high}
configsyncaction: -----> {noaction}
configsyncfilename: ---> {}
configsyncstatus: -----> {syncinitializing}
configsyncuser: -----> {}
configsyncpasswd: -----> ** private **
numshelves: -----> {1}
shelvesarray: -----> {}
numcards: -----> {3}
ipaddress: -----> {0.0.0.0}
alternateipaddress: ---> {0.0.0.0}
countryregion: -----> {us}
primaryclocksource: ---> {0/0/0/0/0}
ringsource: -----> {internalringsourcelabel}
revertiveclocksource: -> {true}
voicebandwidthcheck: --> {false}
alarm-levels-enabled: -> {critical+major+minor+warning}
userauthmode: -----> {local}
radiusauthindex: -----> {0}
secure: -----> {disabled}
webinterface: -----> {enabled}
options: -----> {enablexcardlinkagg}
reservedVlanIdStart: --> {0}
reservedVlanIdCount: --> {0}
snmpVersion: -----> {snmpv2}
persistentLogging: ----> {disabled}

```

5 Create the link aggregation group by adding the links in *active* mode.

Linkagg groups are static and all links in the link aggregation group must be added with the **linkagg add group** command.

```

zSH> linkagg add group 1-a-1-0/linkagg link 1-a-2-0/eth mode active
Interface 1-a-1-0/linkagg does not exist
Link aggregation successfully created.

```

```

zSH> linkagg add group 1-a-1-0/linkagg link 1-a-3-0/eth mode active
Link aggregation successfully created.

```

```

zSH> linkagg add group 1-a-1-0/linkagg link 1-a-4-0/eth mode active
Link aggregation successfully created.

```

```

zSH> linkagg add group 1-a-1-0/linkagg link 1-a-5-0/eth mode active
Link aggregation successfully created.

```

6 Verify that the status of the aggregation mode is *active*.

```

zSH> get ether 1-a-5-0/eth
ether 1-a-5-0/eth
autonegstatus: ----> {enabled}
mauType: -----> {mau1000baselxfid}
restart: -----> {norestart}
ifType: -----> {mau1000baselxfid}
autonegcap: -----> {b10baseTFD+b100baseTXFD+b1000baseTFD}

```

```

remotefault: -----> {noerror}
clksrc: -----> {automatic}
pauseFlowControl: -> {disabled}
aggregationMode: --> {active}
linkStateMirror: --> {0/0/0/0/0}
maxFrameSize: -----> {0}
ingressRate: -----> {0}
ingressBurstSize: -> {0}
egressRate: -----> {0}
egressBurstSize: --> {0}

```

7 View the automatically created link aggregation across uplinks group.

```
zSH> linkagg show
```

LinkAggregations:

slot	unit	ifName	partner: Sys	Pri	grp ID	status	agg mode
a*	1	1-a-1-0	f8:66:f2:0d:3c:41	0xa	0xe	ACT	Active
		links	slot port subport		status		(xcard)
		1-a-2-0	a 2 0		ACT		
		1-a-5-0	a 5 0		ACT		
		1-a-4-0	a 4 0		ACT		
		1-a-3-0	a 3 0		ACT		
b*	1	1-b-1-0	f8:66:f2:0d:3c:41	0xa	0xe	ACT	Active
		links	slot port subport		status		(xcard)
		1-b-2-0	b 2 0		ACT		
		1-b-5-0	b 5 0		ACT		
		1-b-4-0	b 4 0		ACT		
		1-b-3-0	b 3 0		ACT		

global linkagg group red type: xcard

Delete a link aggregation group

Deleting a link aggregation group is the same for link aggregation and link aggregation across cards.

Deleting a cross-card link aggregation group

1 Delete a link from the link aggregation group.

```
zSH> linkagg delete group 1-a-1-0/linkagg link 1-a-2-0/eth
```

Link successfully deleted from aggregation.

Verify the deletion.

```
zSH> linkagg show
```

LinkAggregations:

slot	unit	ifName	partner: Sys	Pri	grp ID	status	agg mode
a*	1	1-a-1-0	f8:66:f2:0d:3c:41	0xa	0xe	ACT	Active
		links	slot port subport		status		(xcard)

```

-----
1-a-5-0      a      5      0      ACT
1-a-4-0      a      4      0      ACT
1-a-3-0      a      3      0      ACT
b* 1 1-b-1-0  f8:66:f2:0d:3c:41  0xa  0xe  ACT  Active
links        slot    port    subport    status    (xcard)
-----
1-b-5-0      b      5      0      ACT
1-b-4-0      b      4      0      ACT
1-b-3-0      b      3      0      ACT
global linkagg group red type: xcard

```

2 Delete all the links in the link aggregation group to delete the group.

```

zSH> linkagg delete group 1-a-1-0/linkagg link 1-a-3-0/eth
Link successfully deleted from aggregation.

```

```

zSH> linkagg delete group 1-a-1-0/linkagg link 1-a-4-0/eth
Link successfully deleted from aggregation.

```

```

zSH> linkagg delete group 1-a-1-0/linkagg link 1-a-5-0/eth
Link successfully deleted from aggregation.

```



Note: If a *linkagg* bridge exists on the physical interface associated with the link aggregation group, you will not be able to delete the links.

Configure link aggregation on Ethernet line card ports

Configuring the MXK to run link aggregation requires setting the mode on the Ethernet port to *active*.

When the mode is *active* the link aggregation group is automatically created and is composed of up to two links depending on the remote device.

Configure line card Ethernet ports for LACP

When the aggregation mode on the Ethernet line card ports is *active*, the device sends and receives LACP and the link aggregation is dynamic, i.e., the link aggregation groups are automatically created.

Enabling LACP on Ethernet line card ports

Enable two or more Ethernet ports on the line card for LACP.

- 1 Connect the MXK to the LACP enabled switch.
- 2 Create the link aggregation group by adding the links in *active* mode.

Linkagg groups are static and all links in the link aggregation group must be added with the **linkagg add group** command.

```
zSH> linkagg add group 1-1-1-0/linkagg link 1-1-1-0/eth mode active
Interface 1-1-1-0/linkagg does not exist
Link aggregation successfully created.
```

```
zSH> linkagg add group 1-1-1-0/linkagg link 1-1-2-0/eth mode active
Link aggregation successfully created.
```

3 View the link aggregation group.

```
zSH> linkagg show
LinkAggregations:
slot unit ifName          partner: Sys          Pri    grp ID  status  agg mode
-----
  1   1   1-1-1-0          00:00:00:00:00:00    0x0    0x0    OOS    Active
      links          slot    port    subport          status
      -----
        1-1-1-0          1      1      0              OOS
        1-1-2-0          1      2      0              OOS
global linkagg group red type: red
```

Configure link aggregation bridges

Bridge interfaces can be added to ports that are a part of link aggregation groups.

Configure link aggregation uplink bridges

Creating an link aggregated uplink bridge

Unlearned traffic received on this interface is forwarded to the external network.

1 To verify link aggregation groups, enter:

```
zSH> linkagg show
LinkAggregations:
slot unit ifName          partner: Sys          Pri    grp ID  status  agg mode
-----
a*  1   1-a-1-0          00:00:00:00:00:00    0x0    0x0    OOS    Active
      links          slot    port    subport          status
      -----
        1-a-6-0          a      6      0              OOS
b   1   1-b-1-0          00:00:00:00:00:00    0x0    0x0    OOS    Active
      links          slot    port    subport          status
      -----
        1-b-6-0          b      6      0              OOS
global linkagg group red type: red
```

2 To create an uplink bridge with link aggregation, enter:

```
zSH> bridge add 1-a-1-0/linkagg uplink vlan 333
Adding bridge on 1-a-1-0/linkagg
Created bridge-interface-record linkagg-a-1-333/bridge
```

Bridge-path added successfully

Verify the bridge.

```
zSH> bridge show vlan 333
```

Type	Orig VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
upl		Tagged 333	1/a/1/0/linkagg	linkagg-a-1-333/bridge	DWN	S	VLAN 333 default

1 Bridge Interfaces displayed

3 To delete a bridge with link aggregation enter:

```
zSH> bridge delete linkagg-a-1-333/bridge vlan 333
```

Bridge-path deleted successfully

linkagg-a-1-333/bridge delete complete

Creating an uplink bridge on an aggregated Ethernet port

If a bridge is created on a link aggregated Ethernet interface on a physical port, a linkagg bridge is automatically created.

Create the uplink bridge.

```
zSH> bridge add 1-a-6-0/eth uplink vlan 444
```

Adding bridge on 1-a-6-0/eth

Created bridge-interface-record linkagg-a-1-444/bridge

Bridge-path added successfully

Since the Ethernet port *1-a-2-0/eth* is part of a link aggregation group, the bridge type is automatically designated *linkagg*.

Verify the *linkagg* bridge.

```
zSH> bridge show vlan 444
```

Type	Orig VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
upl		Tagged 444	1/a/1/0/linkagg	linkagg-a-1-444/bridge	DWN	S	VLAN 444 default

1 Bridge Interfaces displayed

Configure link aggregation line card bridges

Creating a link aggregated bridge on an Ethernet line card

1 Verify the link aggregation group.

```
zSH> linkagg show
```

LinkAggregations:

slot	unit	ifName	partner: Sys	Pri	grp ID	admin	numLinks
6	1	1-6-1-0	00:00:00:00:00:00	0x0	0x0	up	2
		links	slot port subport		admin		
		1-6-1-0	6 1 0		up		
		1-6-2-0	6 2 0		up		

```
global linkagg group red type: red
```

2 Create the bridge on the link aggregation group.

```
zSH> bridge add 1-6-1-0/linkagg downlink vlan 600
Adding bridge on 1-6-1-0/linkagg
Created bridge-interface-record linkagg-6-1/bridge
```

3 View the bridge created on the link aggregation group.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN      Physical      Bridge
St   Table Data
-----
dwn          600      1/6/1/0/linkagg  linkagg-6-1/bridge
UP
1 Bridge Interfaces displayed
```

Deleting a link aggregation bridge

Delete the link aggregation bridge.

```
zSH> bridge delete linkagg-6-1/bridge vlan 600
linkagg-6-1/bridge delete complete
```

Configure a TLS bridge on a link aggregation bridge

If a port is a part of a link aggregation group, the bridge type *linkagg* is assigned to the bridge interface.

Configuring a TSL link aggregation bridge on an Ethernet port

In this case, a TLS bridge is created on an uplink card Ethernet port that is a member of a link aggregation group.

- 1 Create the TLS bridge on an Ethernet port. This Ethernet port is a member of a link aggregation group, therefore the bridge interface is assigned *linkagg* as the bridge type.

```
zSH> bridge add 1-a-7-0/eth tls vlan 888
Adding bridge on 1-a-7-0/eth
Created bridge-interface-record linkagg-a-1/bridge
Bridge-path added successfully
```

2 View the TLS bridge.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN      Physical      Bridge
St   Table Data
-----
```



```

tls                888        1/a/1/0/linkagg        linkagg-a-1/bridge
DWN
1 Bridge Interfaces displayed

```

3 View the TLS bridge path.

```
zSH> bridge-path show
```

VLAN/SLAN	Bridge	Address
888	N/A	VLAN, Age: 3600, MCAST Age: 250, IGMP Query Interval: 0, IGMP DSCP: 0, Flap Mode: Fast

The bridge-path on TLS bridges are on the VLAN ID, not the bridge interface and are created only for the first instance of TLS and VLAN ID.

Configuring a TLS link aggregation bridge on a link aggregation group

In this case, a TLS bridge is created on a link aggregation group comprised of several Ethernet ports.

1 Verify the linkagg group.

```
zSH> linkagg show
```

```

LinkAggregations:
slot unit ifName      partner: Sys      Pri      grp ID status
agg mode
-----
a*  1  1-a-1-0      00:01:47:31:ce:c8  0x8000  0x4000 ACT
Active
      links      slot      port      subport      status
-----
      1-a-4-0      a         4         0             ACT
      1-a-7-0      a         7         0             ACT
      1-a-6-0      a         6         0             ACT
      1-a-5-0      a         5         0             ACT
b   1  1-b-1-0      00:01:47:16:47:a2  0x8000  0x4000 DSA
Active
      links      slot      port      subport      status
-----
      1-b-4-0      b         4         0             DSA
      1-b-7-0      b         7         0             DSA
      1-b-6-0      b         6         0             DSA
      1-b-5-0      b         5         0             DSA
global linkagg group red type: red

```

2 Create a TLS bridge on the linkagg group interface.

```
zSH> bridge add 1-a-1-0/linkagg tls vlan 888
```

```

Adding bridge on 1-a-1-0/linkagg
Created bridge-interface-record linkagg-a-1/bridge
Bridge-path added successfully

```

The bridge-path on TLS bridges are on the VLAN ID, not the bridge interface and are created only for the first instance of TLS and VLAN ID.

3 Verify the bridge.

```
zSH> bridge show
      Orig
Type VLAN/SLAN  VLAN/SLAN      Physical      Bridge
St  Table Data
-----
tls           888          1/a/1/0/linkagg  linkagg-a-1/bridge
DWN
1 Bridge Interfaces displayed
```

4 View the TLS bridge-path.

```
zSH> bridge-path show
VLAN/SLAN  Bridge      Address
-----
      888 N/A          VLAN, Age: 3600, MCAST Age: 250, IGMP
Query Interval: 0, IGMP DSCP: 0, Flap Mode: Fast
```

9

MXK ETHERNET UPLINK CARDS

This chapter describes the MXK 100/1000 Ethernet and 10 GE uplink cards and uplink card configuration:

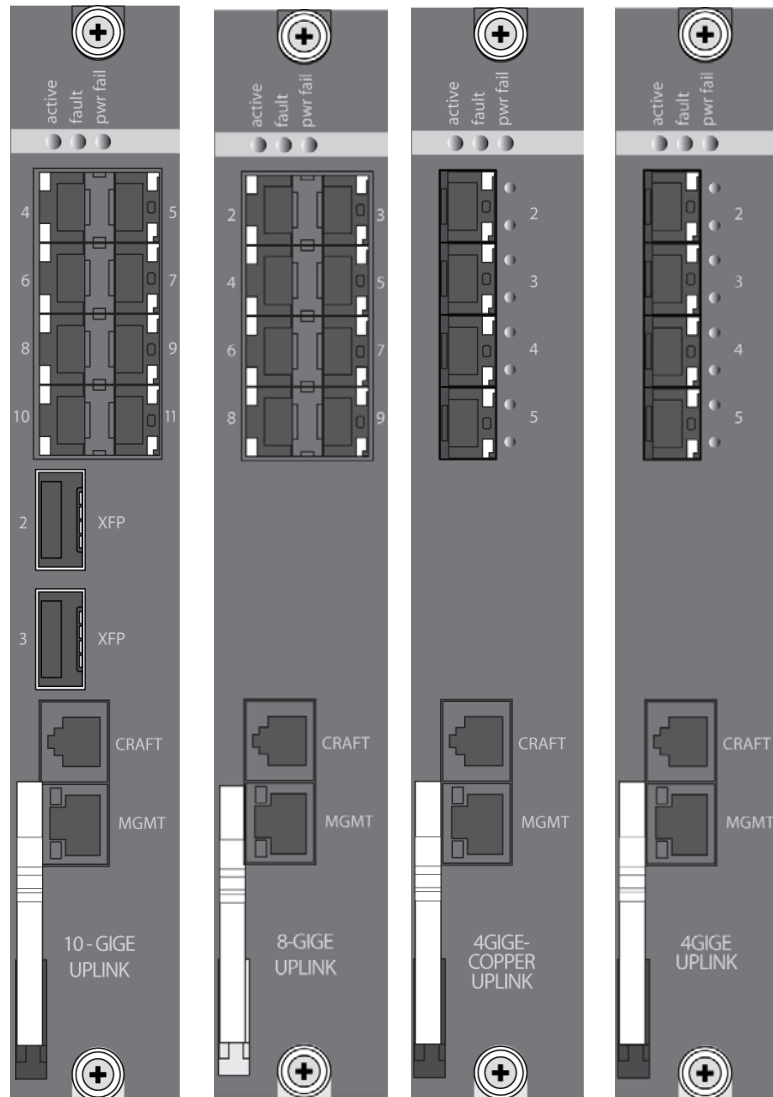
- [MXK 100/1000 Ethernet and 10 GE uplink cards, page 607](#)
- [MXK Ethernet uplink cards with clocking, page 612](#)
- [Equipment protection and facility protection on the MXK, page 626](#)
- [Facility protection on the MXK, page 637](#)
- [EAPS, page 640](#)
- [Displaying and updating Ethernet interfaces, page 668](#)
- [Small form factor pluggables, page 670](#)
- [Uplink card pinouts, page 670](#)

MXK 100/1000 Ethernet and 10 GE uplink cards

This section describes the MXK 100/1000 Ethernet and 10 GE uplink cards and how to configure the cards.

- [MXK 100/1000 Ethernet and 10 GE uplink cards overview, page 608](#)
- [MXK Ethernet uplink card specifications, page 609](#)
- [Disable Tx power on the uplink standby card, page 633](#)
- [View additional card and system information, page 634](#)
- [Displaying and updating Ethernet interfaces, page 668](#)

MXK 100/1000 Ethernet and 10 GE uplink cards overview



The MXK uplink cards are:

- MXK-UPLINK-2X10G-8X1GE
- MXK-UPLINK-8X1GE
- MXK-UPLINK-4X1GE-CU
- MXK-UPLINK-4X1GE

MXK-UPLINK-2X10G-8X1GE

The MXK-UPLINK-2X10G-8X1GE uplink card provides high-speed Gigabit Ethernet interfaces with active/standby redundancy. This uplink card consists of two 10 GE and eight 100/1000 Ethernet interfaces.

MXK MXK-UPLINK-8X1GE

The MXK MXK-UPLINK-8X1GE uplink card provides high-speed Gigabit Ethernet interfaces with active/standby redundancy. This uplink card consists of eight 100/1000 Ethernet interfaces.

MXK-UPLINK-4X1GE-CU

The MXK-UPLINK-4X1GE-CU uplink card provides high-speed Gigabit Ethernet interfaces with active/standby redundancy. This uplink card consists of four 100/1000 Ethernet interfaces and supports only copper line cards.

MXK-UPLINK-4X1G

The MXK-UPLINK-4X1GE uplink card provides high-speed Gigabit Ethernet interfaces with active/standby redundancy. This uplink card consists of four 100/1000 Ethernet interfaces and supports all line cards.

All the uplink cards provide a serial craft interface and a 10/100 Ethernet interface for management. The 100/1000 and 10 GE ports are available for link aggregation.

The 100/1000 Ethernet and 10 GE interfaces support a number of small form factor pluggables (SFPs) and XFPs respectively that enable the card to interface with a variety of media types. (For more information see [Small form factor pluggables on page 670](#).)

MXK Ethernet uplink card specifications

[Table 47](#) provides the MXK-UPLINK-2X10G-8X1GE uplink card specifications.

Table 47: Uplink card MXK-UPLINK-2X10G-8X1GE specifications

Specification	Description
Size	1 slot
Physical interfaces	Two 10 GE ports with XFPs. See Chapter 18, Small Form Factor Pluggable (SFP) Connectors, on page 1117. Eight 100/1000 Ethernet ports with SFPs. The SFPs can be twisted pair 1000baseT or fiber (SX, LX or ZX). See Small form factor pluggables on page 670. The optical interfaces are class 1 Laser International Safety Standard IEC 825 compliant RJ45 Ethernet 10/100 Ethernet interface for management RS232D serial craft interface
Standards supported	AF-PHY-0086.001 G.965 and ETSI EN 300 347-1 V2.2.2 (V5.2) Gigabit Ethernet (GigE) IEEE 802.3

Table 47: Uplink card MXK-UPLINK-2X10G-8X1GE specifications (Continued)

Specification	Description
Management interface	RS-232D serial craft port Management Ethernet 10/100 port routable for out-of-band management. SNMP–None or all.
Redundancy	Card redundancy
Power consumption	Nominal: 88 W 10G ports: 3 W 1G ports: 1.4 W Maximum: 112W

[Table 48](#) provides the MXK-UPLINK-8X1GE uplink card specifications.

Table 48: Uplink card MXK-UPLINK-8X1GE specifications

Specification	Description
Size	1 slot
Physical interfaces	Eight 100/1000 Ethernet ports with SFPs. The SFPs can be twisted pair 1000baseT or fiber (SX, LX or ZX). See Small form factor pluggables on page 670 . The optical interfaces are class 1 Laser International Safety Standard IEC 825 compliant RJ45 Ethernet 10/100 Ethernet interface for management RS232D serial craft interface
Standards supported	AF-PHY-0086.001 G.965 and ETSI EN 300 347-1 V2.2.2 (V5.2) Gigabit Ethernet (GigE) IEEE 802.3
Management interface	RS-232D serial craft port Management Ethernet 10/100 port routable for out-of-band management. SNMP–None or all.
Redundancy	Card redundancy
Power consumption	Nominal: 80 W 1 GE ports: 1.4 W Maximum: 43 W

[Table 49](#) provides the MXK-UPLINK-4X1GE-CU uplink card specifications.

Table 49: Uplink card MXK-UPLINK-4X1GE-CU specifications

Specification	Description
Size	1 slot
Physical interfaces	Four 100/1000 Ethernet ports with SFPs. The SFPs are copper and fiber 100M/1G). See Small form factor pluggables on page 670. The optical interfaces are class 1 Laser International Safety Standard IEC 825 compliant RJ45 Ethernet 10/100 Ethernet interface for management RS232D serial craft interface
Standards supported	AF-PHY-0086.001 G.965 and ETSI EN 300 347-1 V2.2.2 (V5.2) Gigabit Ethernet (GigE) IEEE 802.3
Management interface	RS-232D serial craft port Management Ethernet 10/100 port routable for out-of-band management. SNMP–None or all.
Redundancy	Card redundancy
Power consumption	Nominal: 20W 1 GE ports: 1.4 W Maximum: 36 W

MXK uplink card types

[Table 50](#) provides the card type and software image for the MXK uplink cards.

Table 50: MXK uplink card types

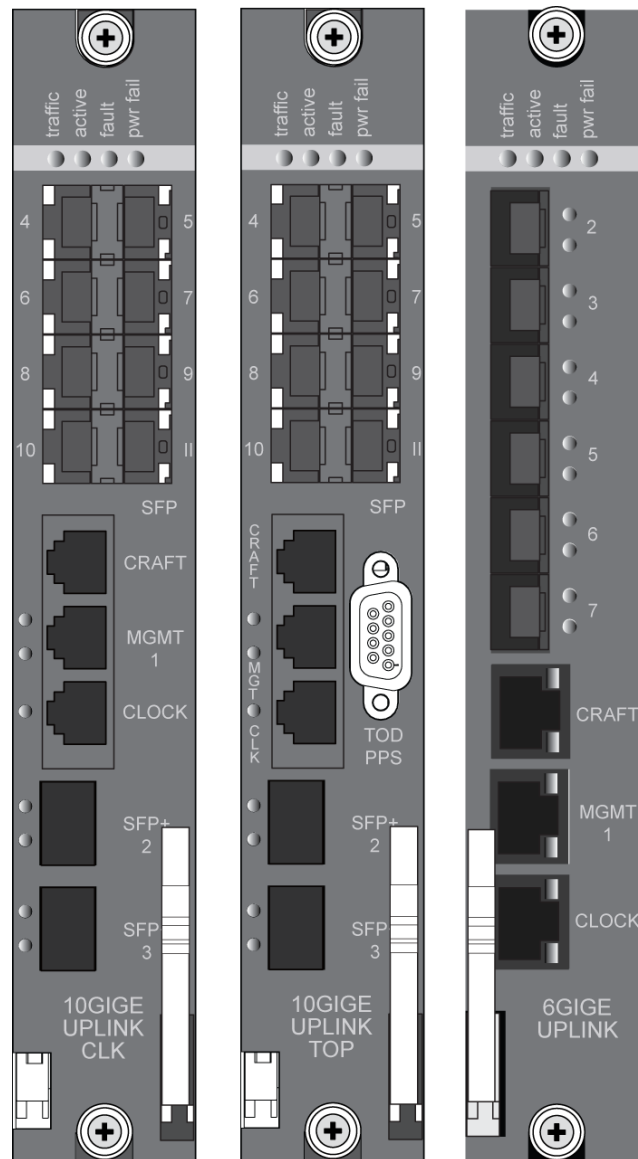
Card	Type	Name of software image
MXK-UPLINK-2X10G-8X1GE	10100	<i>mxup2tg8g.bin</i>
MXK-UPLINK-8X1GE	10101	<i>mxupg8g.bin</i>
MXK-UPLINK-4X1GE-CU	10121	<i>mxup4gcopper.bin</i>
MXK-UPLINK-4X1GE	10125	<i>mxup4g.bin</i>

MXK Ethernet uplink cards with clocking

This section describes the MXK uplink clocking cards with T1/E1 or BITS timing or timing-over-packet (TOP):

- [MXK Ethernet uplink cards with clocking overview, page 613](#)
- [MXK 10-port 2X 10G 8X 1-GE uplink card with Timing over Packet \(TOP\), page 614](#)
- [MXK 10-port 2X 10G 8X 1-GE uplink card with T1/E1 or BITS timing inputs, page 615](#)
- [MXK 6-port 6X 1-GE uplink card with T1/E1 or BITS timing inputs, page 617](#)
- [MXK uplink cards with clocking card types, page 619](#)
- [MXK uplink clocking cards LED redundancy status, page 620](#)
- [MXK Ethernet uplink cards pinouts, page 621](#)
- [Cables and clocking, page 623](#)

MXK Ethernet uplink cards with clocking overview



The MXKuplink cards with timing are:

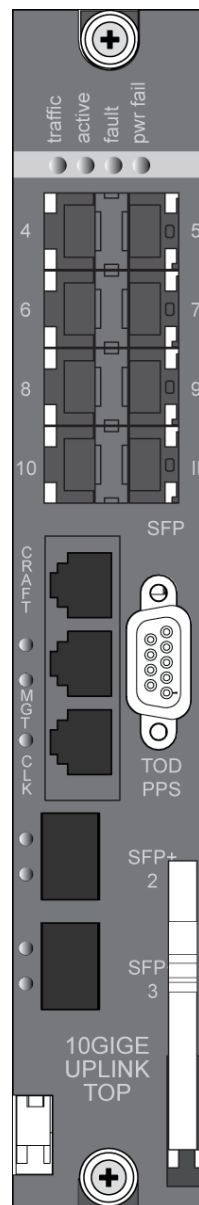
- MXK-UPLINK-2X10G-8X1G-TOP
See [MXK 10-port 2X 10G 8X 1-GE uplink card with Timing over Packet \(TOP\)](#), page 614.
- MXK-UPLINK-2X10G-8X1G-CLK
See [MXK 10-port 2X 10G 8X 1-GE uplink card with T1/E1 or BITS timing inputs](#), page 615.
- MXK-UPLINK-6X1G-CLK

See [MXK 6-port 6X 1-GE uplink card with T1/E1 or BITS timing inputs](#), page 617.

MXK 10-port 2X 10G 8X 1-GE uplink card with Timing over Packet (TOP)

This section describes:

- [10-port 2X 10G 8X 1-GE uplink card \(TOP\) overview](#), page 614
- [MXK-UPLINK-2X10G-8X1G-TOP card specifications](#), page 615



10-port 2X 10G 8X 1-GE uplink card (TOP) overview

The MXK-UPLINK-2X10G-8X1G-TOP uplink card provides high-speed Gigabit Ethernet interfaces with active/standby redundancy. This uplink card consists of two 10 GE and eight 100/1000 Ethernet interfaces and supports all line cards.

The CLOCK input port accepts T1/E1 or BITS timing.

The MXK-UPLINK-2X10G-8X1G-TOP uplink card also supports Precision Time Protocol (PTP) and 1599v2 and SyncE clocking. For information on clocking configuration, see [MXK Clocking](#), page 173.

MXK-UPLINK-2X10G-8X1G-TOP card specifications

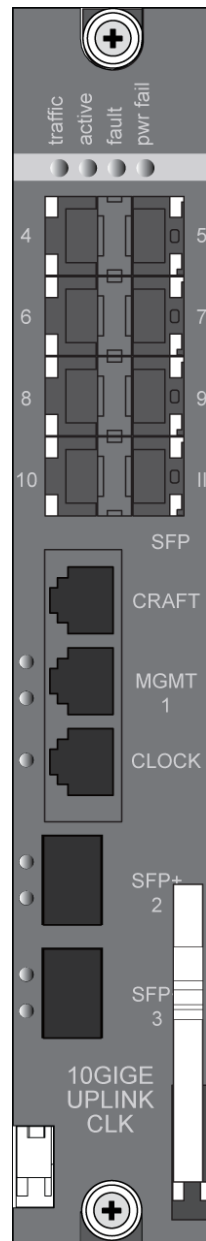
Table 51: Uplink card MXK-UPLINK-2X10G-8X1G-TOP specifications

Specification	Description
Size	1 slot
Physical interfaces	Eight 100/1000 Ethernet ports with SFPs. The SFPs can be twisted pair 1000baseT or fiber (SX, LX or ZX). See Small form factor pluggables on page 670 Two 10 GE ports with SFP+. See Chapter 18, Small Form Factor Pluggable (SFP) Connectors, on page 1117 The optical interfaces are class 1 Laser International Safety Standard IEC 825 compliant RJ45 Ethernet 10/100 Ethernet interface for management RS232D serial craft interface RJ45 accepts T1/E1 and BITS timing DE-9S connector for PPS and TOD input/output
Standards supported	Bridging 802.1D VLAN 802.1Q with 802.1P Multicast IGMP v2 and v3 proxy ITU G.984.1-984.4 OMCI IEEE 802.ah IEEE 802.3ad LACP DHCP relay 802.1 ag 1588v2 SyncE
Management interface	RS-232D serial craft port Management Ethernet 10/100 port routable for out-of-band management. SNMP—None or all.
Redundancy	Card redundancy
Power consumption	Nominal: 100 W 10G ports: 3.0 W 1GE ports: 1.4 W Maximum: 124 W

MXK 10-port 2X 10G 8X 1-GE uplink card with T1/E1 or BITS timing inputs

This section describes:

- [MXK 10-port 2X 10G 8X 1-GE uplink card with T1/E1 or BITS timing inputs overview, page 616](#)
- [MXK-UPLINK-2X10G-8X1G-CLK card specifications, page 617](#)



MXK 10-port 2X 10G 8X 1-GE uplink card with T1/E1 or BITS timing inputs overview

The MXK-UPLINK-2X10G-8X1G-CLK uplink card provides high-speed Gigabit Ethernet interfaces with active/standby redundancy. This uplink card consists of two 10 GE and eight 100/1000 Ethernet interfaces and supports all line cards.

The CLOCK input port accepts T1/E1 or BITS timing. For information on clocking configuration, see [MXK Clocking, page 173](#).

MXK-UPLINK-2X10G-8X1G-CLK card specifications

Table 52: Uplink card MXK-UPLINK-2X10G-8X1G-TOP specifications

Specification	Description
Size	1 slot
Physical interfaces	Eight 100/1000 Ethernet ports with SFPs. The SFPs can be twisted pair 1000baseT or fiber (SX, LX or ZX). See Small form factor pluggables on page 670 Two 10 GE ports with SFP+. See Chapter 18, Small Form Factor Pluggable (SFP) Connectors, on page 1117 The optical interfaces are class 1 Laser International Safety Standard IEC 825 compliant RJ45 Ethernet 10/100 Ethernet interface for management RS232D serial craft interface RJ45 accepts T1/E1 and BITS DE-9S connector for PPS and TOD input/output
Standards supported	Bridging 802.1D VLAN 802.1Q with 802.1P Multicast IGMP v2 and v3 proxy ITU G.984.1-984.4 OMCI IEEE 802.ah IEEE 802.3ad LACP DHCP relay 802.1 ag
Management interface	RS-232D serial craft port Management Ethernet 10/100 port routable for out-of-band management. SNMP—None or all.
Redundancy	Card redundancy
Power consumption	Nominal: 100 W 10G ports: 3.0 W 1GE ports: 1.4 W Maximum: 124 W

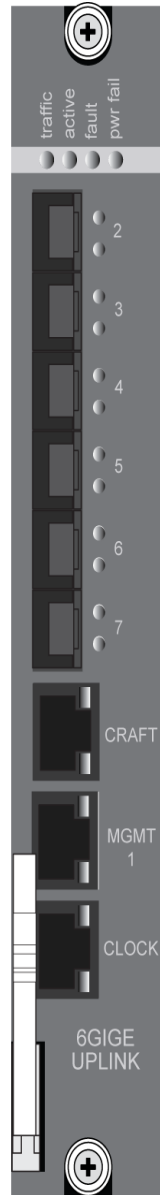
MXK 6-port 6X 1-GE uplink card with T1/E1 or BITS timing inputs

This section describes:

- [MXK 6-port 6X 1-GE uplink card with T1/E1 or BITS timing inputs overview, page 618](#)

- [MXK 6-port 6X 1-GE uplink card with T1/E1 or BITS timing inputs specifications, page 619](#)

MXK 6-port 6X 1-GE uplink card with T1/E1 or BITS timing inputs overview



The MXK-UPLINK-6X1G-CLK uplink card provides high-speed Gigabit Ethernet interfaces with active/standby redundancy. This uplink card consists of six 100/1000 Ethernet interfaces and supports all line cards. The CLOCK input port supports TI/E1 or BITS.

MXK 6-port 6X 1-GE uplink card with T1/E1 or BITS timing inputs specifications

Table 53: Uplink card MXK-UPLINK-6X1GE specifications

Specification	Description
Size	1 slot
Physical interfaces	Six 100/1000 Ethernet ports with SFPs. The SFPs can be twisted pair 1000baseT or fiber (SX, LX or ZX). See Small form factor pluggables on page 670 The optical interfaces are class 1 Laser International Safety Standard IEC 825 compliant RJ45 Ethernet 10/100 Ethernet interface for management RS232D serial craft interface RJ45 accepts T1/E1 and BITS
Standards supported	Bridging 802.1D VLAN 802.1Q with 802.1P Multicast IGMP v2 and v3 proxy ITU G.984.1-984.4 OMCI IEEE 802.ah IEEE 802.3ad LACP DHCP relay 802.1 ag
Management interface	RS-232D serial craft port Management Ethernet 10/100 port routable for out-of-band management. SNMP–None or all.
Redundancy	Card redundancy
Power consumption	Nominal: 25 W 1 GE ports: 1.4 W Total Max: 43 W

MXK uplink cards with clocking card types

[Table 54](#) provides the card type and software image for the MXK uplink cards with clocking.

Table 54: MXK uplink cards with clocking card types

Card	Type	Software image
MXK-UPLINK-6X1GE-CLK	10130	<i>mxup6g.bin</i>

Table 54: MXK uplink cards with clocking card types (Continued)

Card	Type	Software image
MXK-UPLINK-2X10G-8X1GE-CLK	10150	<i>mxup2tg8gtop.bin</i>
MXK-UPLINK-2X10G-8X1GE-TOP	10150	<i>mxup2tg8gtop.bin</i>

MXK uplink clocking cards LED redundancy status

Uplink cards have LEDs which illuminate to indicate their redundancy status. A solid green LED indicates the card is active, a blinking green LED indicates the card is standby.

[Table 55](#) describes the LEDs on the MXK uplink cards with clocking.

Table 55: MXK uplink card LED indications

LED	Description
Traffic	ON: The Traffic LED indicates the Active card is receiving traffic from the network on one or more of the uplink ports. OFF: The Active card is not receiving traffic from the network.
Active (Green)	Active uplink card: The Active LED blinks (2 Hz) during POST then stops blinking and remains ON after booting up (approximately five minutes). Standby uplink card: Slowly blinks indefinitely, 1/2 to 1 Hz indicating redundancy ready.
Fault (Yellow)	This LED is ON when: The card is booting. The card detected a hardware failure or the card is not provisioned. If the LED is ON for a provisioned card, the card need to be repaired. Standby uplink card only: When the Active light is slowly blinking, which means that the standby card is UP and Redundancy Ready, the yellow fault light indicates a major alarm on one of the Standby card's Ethernet ports, such as link down. This situation does not indicate a hardware fault. When the Active light is OFF or Blinking Fast, the yellow Fault light indicates the card is not provisioned or has not finished booting to the Redundancy Ready state.
Pwr Fail	ON: The card has detected a local on-board power failure. While the card may operate properly, it needs repair as soon as possible. For System power status, refer to the appropriate chassis LEDs.

Table 55: MXK uplink card LED indications (Continued)

LED	Description
CLOCK RJ-45 LEDs	CLOCK RJ-45 green LED.  Note: The operation of this green LED does not represent DS1 port status. The unique meaning of this LED is described below. T1/E1 clocking: After installing a T1/E1 cable, when a valid clock source is present, the LED light is green whether or not this is the clock source selected by the system clock manager. It may take up to 30 seconds for the system to detect that this clock is valid. The light is OFF when a valid clock source is not present. T1/E1 does not receive data, just clock, therefore it is the same as LINK. BITS clocking: After installing a BITS cable, if the system clock manager selects BITS as the clock source and the PLL is locked to this source confirming it is a valid clock source, the LED light is green. See Chapter 3, MXK Clocking for how to set T1/E1 clocking.

MXK Ethernet uplink cards pinouts

This section lists the pinouts for the following interfaces that are found on uplink cards:

- [Ethernet port pinouts, page 621](#)
- [Clocking port pinouts, page 622](#)
- [Serial \(craft\) port pinouts, page 622](#)

Ethernet port pinouts

[Table 56](#) lists the standard Ethernet port pinouts on uplink cards.

Table 56: Uplink card standard Ethernet port pinouts

Pin	Function
1	Tx +
2	Tx -
3	Rx +
4	Not used
5	Not used
6	Rx -

Table 56: Uplink card standard Ethernet port pinouts (Continued)

Pin	Function
7	Not used
8	Not used

Clocking port pinouts

[Table 57](#) lists the pinouts for the uplink card clock port. Pinouts follow the standard specifications with pins 1 and 2 for receive and pins 4 and 5 for transmit. Pins 6, 7, and 8 are used for 2.048 MHz square wave signals for BITS.

*Connect BITS select to ground to use BITS clock input.

Table 57: Standard cable pinouts for clock types BITS and T1/E1

Pin	BITS	T1/E1
1	Not connected	Rx ring
2	Not connected	Rx tip
3	Not connected	Not connected
4	Not connected	Tx Ring
5	Not connected	Tx Tip
6	BITS select connected to pin 8	Not connected
7	BITS clock	Not connected
8	Ground	Not connected

Serial (craft) port pinouts

[Table 58](#) lists the uplink cards' serial (craft) port pinouts. The serial (craft) port uses RS232 signal levels and is configured as DCE.



Note: Normally, only pins 4 (SGND), 5 (RD) and 6 (TD) need to be connected to the external DTE (terminal or computer). The uplink provides constant ON-level (positive) levels on output pins 1 (DSR), 2 (DCD) and 7 (CTS) in case they are needed by the DTE. Input signals from the DTE may be connected to pins 3 (DTR) and 8 (RTS), but the uplink does not require or examine them.

[Table 58](#) lists the pinouts to connect a DE-9S connector to the MXK RJ45 serial craft port.

Table 58: Craft RJ45 to DE-9S adapter pinouts

RJ-45 pin	Color	Function	DE-9S pin
1	N/A	DCE Ready, Ring Indicator (DSR/RI)	not used
2	N/A	Received Line Signal Detector (DCD)	not used
3	N/A	DTE Ready (DTR)	not used
4	Red	Signal Ground (SGND)	5
5	Green	Received Data (RD)	2
6	Yellow	Transmitted Data (TD)	3
7	N/A	Clear To Send (CTS)	Looped to pin 8
8	N/A	Request To Send (RTS)	Looped to pin 7

Cables and clocking

For information on clocking configuration, see [MXK Clocking, page 173](#).

The MXK uplink cards provide clock input ports to connect a T1/E1 or BITS external clock reference. Figures [Figure 86](#), [Figure 87](#) and [Figure 88](#) show the cabling for the MXK-UPLINK-6X1G-CLK, MXK-UPLINK-2X10G-8X1GE-CLK, and MXK-UPLINK-2X10G-8X1GE-TOP, respectively.

Figure 86: Cabling for the CLOCK port on MXK-UPLINK-6X1G-CLK

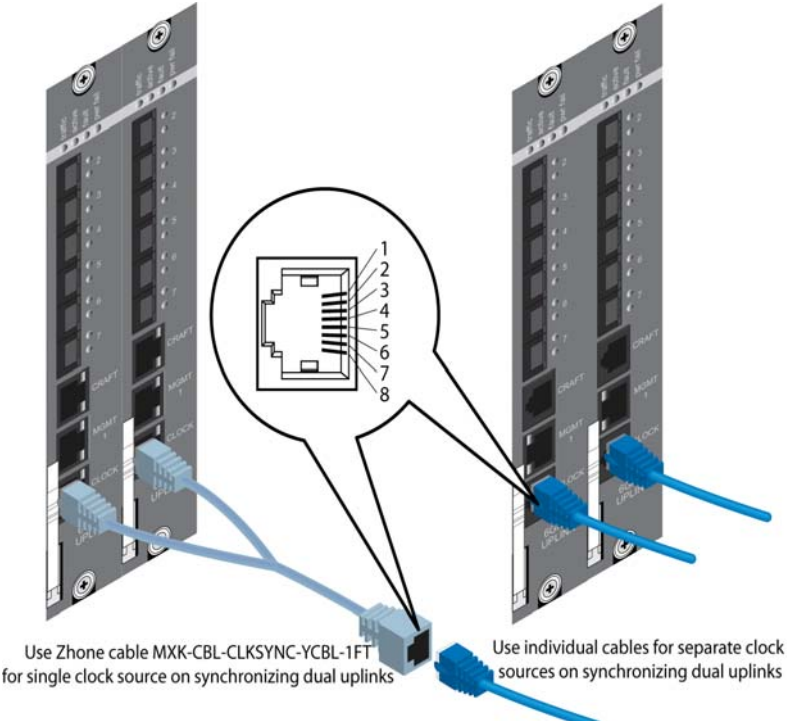


Figure 87: Cabling for the CLOCK port on MXK-UPLINK-2X10G-8X1GE-CLK

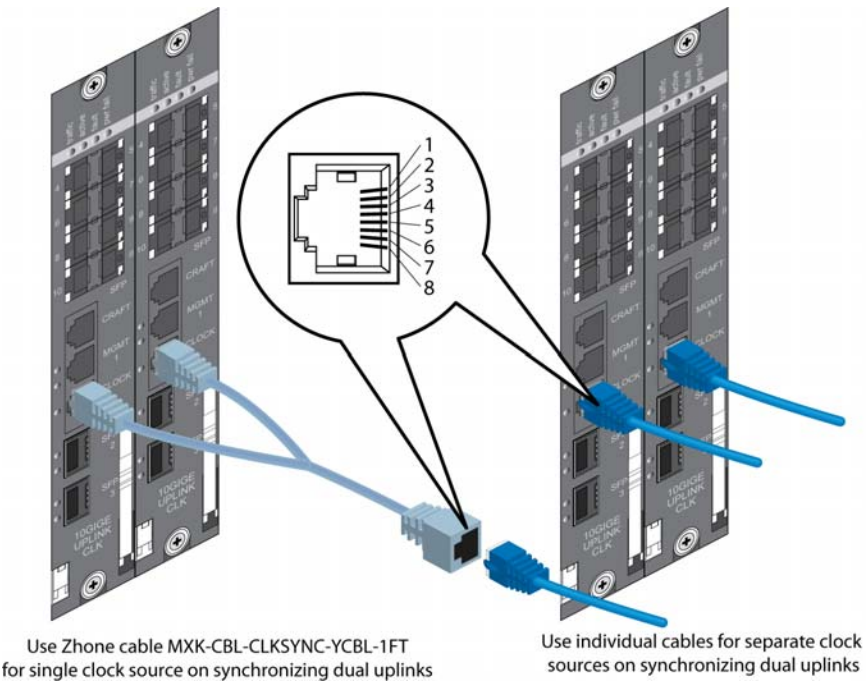
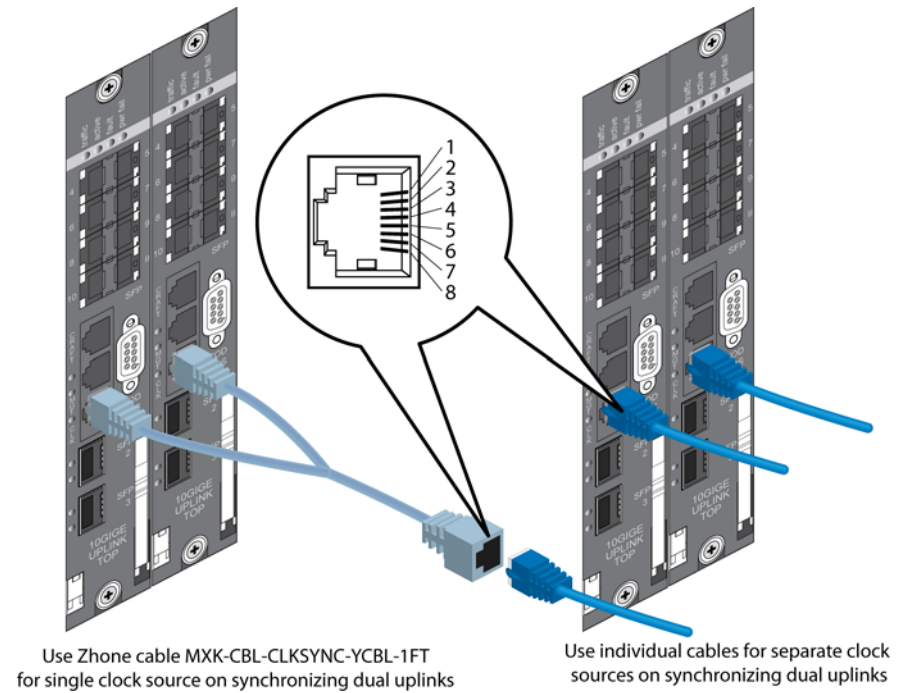


Figure 88: Cabling for the CLOCK port on MXK-UPLINK-2X10G-8X1GE-TOP

Equipment protection and facility protection on the MXK

This section describes:

- [MXK redundant uplinks for equipment protection configuration, page 626](#)
- [Disable Tx power on the uplink standby card, page 633](#)
- [View additional card and system information, page 634](#)
- [MXK facility protection on uplink cards \(2.1.3\), page 634](#)
- [Configure line-red uplink ports for concurrent EAPS \(2.2.x\), page 635](#)

For MXK 2.1.3 and earlier, those redundant uplink cards that share the same line group number and are configured with line-red interface/type timeout 0 a link down will cause the Active card to reboot and transfer traffic from the Active card to the Standby card. See

For MXK 2.2, those redundant uplink cards that share the same line group number, the default for all uplink ports is line-red timeout 0. Now, when a link down occurs, a switchover from the Active port to the Standby port occurs without the Active uplink card rebooting.

MXK redundant uplinks for equipment protection configuration

This section describes how to configure a redundant uplink card for equipment protection on the MXK. After insertion, the uplink card in slot *a* automatically configures when the software image is on the flash.



Note: MXK uplink cards must be installed in the middle slots a and b of the MXK chassis.

[Table 59](#) provides the type and software image for the uplink cards on the MXK. The parameters for the software image of the card type are found in the **card-profile** for that software image.

Table 59: MXK uplink card types

Card	Type	Name of software image
MXK-UPLINK-2X10G-8X1GE	10100	<i>mxup2tg8g.bin</i>
MXK-UPLINK-8X1GE	10101	<i>mxupg8g.bin</i>
MXK-UPLINK-4X1GE-CU	10121	<i>mxup4gcopper.bin</i>
MXK-UPLINK-4X1GE	10125	<i>mxup4g.bin</i>
MXK-UPLINK-6X1G-CLK	10130	<i>mxup6g.bin</i>
MXK-UPLINK-2X10G-8X1G-TOP	10150	<i>mxup2tg8gtop.bin</i>
MXK-UPLINK-2X10G-8X1G-CLK	10150	<i>mxup2tg8gtop.bin</i>

**Caution:**

You must configure redundant physical interfaces on *both* the active and standby cards. In addition, you must manually keep the configuration of the physical interfaces on the active and standby cards in sync.

Each uplink card must run the same software version and have the same size flash card.



Note: When configuring the redundant uplink card, the settings in the **card-profile** for the both cards must be identical.

Configuring redundant uplink cards (without clocking)

For equipment protection on the MXK, the uplink cards in slot *a* and slot *b* must be redundant. This means that if the Active card goes down, the Active card reboots and all traffic is switched to the Standby card which then becomes the Active card.

To configure redundant uplink cards on the MXK:

- 1 Insert the active uplink card in slot *a*.
- 2 Insert the redundant uplink card in slot *b*.

The group ID of the **card-profile** for slot *b* must match the group ID of the **card-profile** of slot *a*.

To modify a parameter in the **card-profile**, use the **update card-profile shelf/slot/card** type command.

- 3 Verify the state of the uplink card in slot *b*.

```
zSH> slots
MXK 819
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (NOT_PROV)
Cards
  1: TAC ITM RING (NOT_PROV)
  4: MXK 4 PORT GPON (RUNNING)
  5: MXK 8 PORT GPON (CONFIGURING)
  6: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
  7: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
  8: MXK ADSL-48-A Bonded/with 600 Ohm Splitter (CONFIGURING)
 10: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (NOT_PROV)
 12: MXK 24 PORT VDSL2 (RUNNING)
 13: MXK 20 ACT ETH (RUNNING)
```

- 4 Verify the group ID of the uplink card in slot *a*.

```
zSH> get card-profile 1/a/10100
card-profile 1/a/10100
sw-file-name: -----> {mxup2tg8g.bin}
```

```

admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {1}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {unknowntype}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}

```

- 5 Change the group ID in the **card-profile** of slot *b* by entering the **card add** command specifying slot *b* and *group*.

```

zSH> card add b group 1
new card-profile 1/b/10100 added, sw-file-name "mxup2tg8g.bin", 1 option:
card-group-id 1

```

The system updates the **sw-file-name** to match the **card-profile** of slot *a* and changes the **card-group-id** parameter to *1*.

- 6 Verify the **sw-file-name** and **card-group-id** parameters in the **card-profile** of the stand-by uplink card in slot *b*.

```

zSH> get card-profile 1/b/10100
card-profile 1/b/10100
sw-file-name: -----> {mxup2tg8g.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {1}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {unknowntype}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}

```

- 7 To view the status of both uplink cards enter **slots**:

```

zSH> slots

```


MXK 819

Uplinks

a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC))

b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)

Cards

1: TAC ITM RING (NOT_PROV)

4: MXK 4 PORT GPON (RUNNING)

5: MXK 8 PORT GPON (RUNNING)

6: MXK 20 ACT ETH SINGLE SLOT (RUNNING)

7: MXK GSHDSL-24 Bonded/with NTP (RUNNING)

8: MXK ADSL-48-A Bonded/with 600 Ohm Splitter (RUNNING)

10: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (NOT_PROV)

12: MXK 24 PORT VDSL2 (RUNNING)

13: MXK 20 ACT ETH (RUNNING)

- 8** To view card information including the state of the card and how long the card has been running, enter **slots** and specify the slot number of the card:

zSH> **slots a**

MXK 819

Type : *MXK TWO TENGIGE EIGHT GIGE

Card Version : 800-02485-01-A

EEPROM Version : 1

Serial # : 1769060

CLEI Code : No CLEI

Card-Profile ID : 1/a/10100

Shelf : 1

Slot : a

ROM Version : MXK 2.0.100

Software Version: MXK 2.2.1.003

State : RUNNING

Mode : FUNCTIONAL

Heartbeat check : enabled

Heartbeat last : THU JAN 27 15:25:24 2011

Heartbeat resp : 1323

Heartbeat late : 0

Hbeat seq error : 0

Hbeat longest : 9

Fault reset : enabled

Power fault mon : not supported

Uptime : 22 minutes

zSH> **slots b**

MXK 819

Type : MXK TWO TENGIGE EIGHT GIGE

Card Version : 800-02485-01-A

EEPROM Version : 1

Serial # : 1360640

CLEI Code : No CLEI

Card-Profile ID : 1/b/10100

Shelf : 1

Slot : b

ROM Version : MXK 2.0.100

Software Version: MXK 2.2.1.003

```

State           : RUNNING
Mode            : FUNCTIONAL
Heartbeat check : enabled
Heartbeat last  : THU JAN 27 15:27:40 2011
Heartbeat resp  : 154
Heartbeat late  : 0
Hbeat seq error : 0
Hbeat longest   : 4
Fault reset     : enabled
Power fault mon : not supported
Uptime          : 2 minutes

```

9 To view redundancy information, enter the following **showredundancy** commands.

```
zSH> showredundancy
```

```

Redundancy status for card 01: a - Safe, all services have redundant peers
01: a is active storage
01: b is standby storage

```

```
zSH> showredundancy -d
```

```
Redundancy status for card 01: a -
```

Taskname	Active Addr	Standby Addr	Stdby Ready?
=====	=====	=====	=====
InfoServer	01:a:02	01:b:02	Yes
RdsServer	01:a:03	01:b:03	Yes
tNumSrv	01:a:1041	01:b:1030	Yes
tShelfRR	01:a:1042	01:b:1031	Yes
tMAXTask	01:a:1043	01:b:1032	Yes
trapSrv	01:a:25	01:b:25	Yes
tFTD	01:a:67	01:b:67	Yes
TadSrvTask	01:a:1045	01:b:1034	Yes
zCardRed	01:a:26	01:b:26	Yes
ifcftgtask	01:a:78	01:b:78	Yes
L-RR-1/a	01:a:79	01:b:79	Yes
Ccrr-1/30	01:a:64	01:b:64	Yes
MPRR-1/30	01:a:1049	01:b:1037	Yes
CTRR-1/30	01:a:1050	01:b:1038	Yes
VoiceCallSup	01:a:1051	01:b:1039	Yes
LogServer	01:a:08	01:b:08	Yes
NpRedSrv	01:a:58	01:b:58	Yes
_RedSpawnSvrTask	01:a:1055	01:b:1041	Yes
tBondRR	01:a:87	01:b:87	Yes
connmgr	01:a:16	01:b:16	Yes
tIPSLM	01:a:75	01:b:75	Yes
DhcpServerTask	01:a:90	01:b:90	Yes
tEtherOamRp	01:a:83	01:b:83	Yes
tBridgeRP	01:a:65	01:b:65	Yes
filterupdate	01:a:1088	01:b:1057	Yes
RtpMgr	01:a:47	01:b:47	Yes
Safe, all services have redundant peers			
01: a is active storage			
01: b is standby storage			

Configuring redundant uplink cards (with clocking)

- 1 Insert the active uplink card in slot *a*.
- 2 Insert the redundant uplink card in slot *b*.

The group ID of the **card-profile** for slot *b* must match the group ID of the **card-profile** of slot *a*.

To modify a parameter in the **card-profile**, use the **update card-profile shelf/slot/card** type command.

- 3 Verify the state of the uplink card in slot *b*.

```
zSH> slots
MXK 319
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE CLK/with Timing Over Packet (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE CLK/with Timing Over Packet (NOT_PROV)
Cards
  1: MXK 2 10G 8 1G ACT ETH (NOT_PROV)
  6: MXK 8 PORT GPON (NOT_PROV)
```

- 4 Verify the **card-group-id** and the **card-line-type** of the uplink card in slot *a*.

```
zSH> get card-profile 1/a/10150
card-profile 1/a/10150
sw-file-name: -----> {mxup2tg8gtop.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {1}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {ds1}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}
```

- 5 Change the group ID in the **card-profile** of slot *b* by entering the **card add** command specifying slot *b*, *group* and *linetype*.

```
zSH> card add b group 1 linetype ds1
newcard-profile1/b/10150added, sw-file-name "mxup2tg8gtop.bin", 2options: c
ard-group-id 1 card-line-type ds1
```

The system updates the **sw-file-name** to match the **card-profile** of slot *a* and changes the **card-group-id** parameter to *1* and the **card-line-type** to *ds1*.

6 Verify parameters in the **card-profile** of the stand-by uplink card in slot *b*.

```
zSH> get card-profile 1/b/10150
card-profile 1/b/10150
sw-file-name: -----> {mxup2tg8gtop.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {1}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {ds1}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}
```

7 To view the status of both uplink cards enter **slots**:

```
zSH> slots
MXK 319
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE CLK/with Timing Over Packet (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE CLK/with Timing Over Packet (RUNNING)
Cards
  1: MXK 2 10G 8 1G ACT ETH (RUNNING)
  6: MXK 8 PORT GPON (RUNNING)
```

8 To view redundancy information, enter the following **showredundancy** commands.

```
zSH> showredundancy
Redundancy status for card 01: a - Safe, all services have redundant peers
01: a is active storage
01: b is standby storage
```

```
zSH> showredundancy -d
Redundancy status for card 01: a -
```

Taskname	Active Addr	Standby Addr	Stdby Ready?
=====	=====	=====	=====
InfoServer	01:a:02	01:b:02	Yes
RdsServer	01:a:03	01:b:03	Yes
tNumSrv	01:a:1041	01:b:1030	Yes
tShelfRR	01:a:1042	01:b:1031	Yes
tMAXTask	01:a:1043	01:b:1032	Yes

trapSrv	01:a:25	01:b:25	Yes
tFTD	01:a:67	01:b:67	Yes
TadSrvTask	01:a:1045	01:b:1034	Yes
zCardRed	01:a:26	01:b:26	Yes
ifcftgtask	01:a:78	01:b:78	Yes
L-RR-1/a	01:a:79	01:b:79	Yes
Ccrr-1/30	01:a:64	01:b:64	Yes
MPRR-1/30	01:a:1049	01:b:1037	Yes
CTRR-1/30	01:a:1050	01:b:1038	Yes
VoiceCallSup	01:a:1051	01:b:1039	Yes
LogServer	01:a:08	01:b:08	Yes
NpRedSrv	01:a:58	01:b:58	Yes
_RedSpawnSvrTask	01:a:1055	01:b:1041	Yes
tBondRR	01:a:87	01:b:87	Yes
connmgr	01:a:16	01:b:16	Yes
tIPSLM	01:a:75	01:b:75	Yes
DhcpServerTask	01:a:90	01:b:90	Yes
tEtherOamRp	01:a:83	01:b:83	Yes
tBridgeRP	01:a:65	01:b:65	Yes
filterupdate	01:a:1088	01:b:1057	Yes
RtpMgr	01:a:47	01:b:47	Yes

Safe, all services have redundant peers

01: a is active storage

01: b is standby storage

Disable Tx power on the uplink standby card

The **line-red set** command can be used to disable the laser on the redundant uplink card when using a fiber optic SFP used to configure Virtual Router Redundancy Protocol (VRRP).



Note: After a fail over, there will be additional latency to bring the laser back up.

Disabling the Tx power on the uplink standby card

- 1 Disable the laser on the MXK standby card.

```
zSH> line-red set 1-a-2-0/eth standbytx disable
```

- 2 View the *standbytx* status.

```
zSH> line-red show 1-a-2-0/eth
```

redundancy status for 1-a-2-0/eth:

```

      NOREBOOT standbytx DISABLE timeout 0 NONREVERTIVE revert timeout 0
Interface-Type      Interface-Name      Oper-State Oper-Status
=====
Primary            1-a-2-0/eth      Active      UP
Secondary          1-b-2-0/eth      Standby     OOS
```

View additional card and system information

View the EPROM version of the card:

```
zSH> eeshow card a
EEPROM contents: for slot 30
EEPROM_ID    : 00 -- CARD
Version      : 01
Size         : 054
CardType     : 10100 -- MXUP2TG8G
CardVersion  : 800-02485-01-A
SerialNum    : 01769060
ShelfNumber  : 00001
CLEI Code    : No CLEI
Cksum       : 0x8596
  MAC Address are enabled for 11 address(es) starting at:
00:01:47:1A:FE:64
```

View the ROM version of the card:

```
zSH> romversion a
MXK 1.14.3.100
May 20 2008, 21:44:05
```

Verify the existence of a daughter card on the system:

```
zSH> eeshow 1d
EEPROM contents: for slot 30
EEPROM_ID    : 01 -- 1DAUGHTER
Version      : 01
Size         : 022
CardType     : 10100 -- MXUP2TG8G
CardVersion  : 800-02482-01-B
SerialNum    : 01768971
ShelfNumber  : 00001
CLEI Code    : No CLEI
Cksum       : 0xFA1F
```

Verify the existence of a second daughter card by entering **eeshow 2d**.

View the version of the system software:

```
zSH> swversion
Zhone mxUp2Tg8g software version MXK 2.2.1.203
```

MXK facility protection on uplink cards (2.1.3)

When the uplink cards in slot *a* and slot *b* are redundant, you can configure a port on the uplink card to switch from the active port to the standby port when a link fails. This is recommended for any critical link that goes off the device. You use the **line-red set** command to configure a port for switchover.



Note: On the physical card, the green light will blink when traffic has been switched from an active to a standby port. In this case, traffic will be running on both uplink cards in slots *a* and *b*. If a card needs to be pulled, traffic must be switched to one card.

Configuring an uplink port for redundancy

When a port is configured for link redundancy with the **line-red set** command and a link on the port fails, traffic is switched from the active port to the standby port without a reboot on line down.

- 1 Verify any existing redundancy with the **line-red show all** command.

```
zSH> line-red show all
No line redundancy interfaces for system
```

- 2 Create redundancy on a port, in this case port 3.

```
zSH> line-red set 1-a-3-0/eth timeout 0
```

If a link fails on port 3, a failover will now be triggered.

- 3 Verify the redundancy.

```
zSH> line-red show 1-a-3-0/eth
redundancy status for 1-a-3-0/eth:
      REBOOT standbytx ENABLE timeout 0 NONREVERTIVE revert timeout 0
Interface-Type      Interface-Name      Oper-State Oper-Status
=====
Primary            1-a-3-0/eth            Active     UP
Secondary          1-b-3-0/eth            Standby    Trfc-Disable
```

Configure line-red uplink ports for concurrent EAPS (2.2.x)

With the MXK release 2.2.x, all uplink ports default to a line-red state when uplink cards reside in slots *a* and *b* and share the same line group number. When a link down occurs on a port, traffic switches from the Active port to the Standby port.

For concurrent uplinks that will be part of a EAPS ring, the line-red state on the uplink port must be broken so that traffic can be passed on Standby as well as Active. Concurrent uplinks provides the ability for non-active ports to pass traffic.



Note: On the physical card, the green light will blink when traffic has been switched from an active to a standby port. In this case, traffic will be running on uplink cards in both slots *a* and *b*. If an uplink card needs to be pulled, traffic must be switched to one card.

Changing an uplink port from line-red to concurrent for EAPS ring configuration

When a port supports facility redundancy with the **line-red set** command and a link on the port fails, traffic is switched from the active port to the standby port without a reboot on line down.

- 1 Verify the state of the uplink ports with the **line-red show all** command.

```
zSH> line-red show all
```

Line Redundant interfaces for system

Interface-Type	Interface-Name	Oper-State	Oper-Status
=====	=====	=====	=====
Primary	1-a-1-0/eth	Active	UP
Primary	1-b-1-0/eth	Standby	Trfc-Disable
Primary	1-a-2-0/eth	Inactive	Inactive
Primary	1-b-2-0/eth	Inactive	Inactive
Primary	1-a-3-0/eth	Inactive	Inactive
Primary	1-b-3-0/eth	Inactive	Inactive
Primary	1-a-4-0/eth	Inactive	Inactive
Primary	1-b-4-0/eth	Inactive	Inactive
Primary	1-a-5-0/eth	Active	UP
Primary	1-b-5-0/eth	Standby	OOS
Primary	1-a-6-0/eth	Inactive	Inactive
Primary	1-b-6-0/eth	Inactive	Inactive
Primary	1-a-7-0/eth	Inactive	Inactive
Primary	1-b-7-0/eth	Inactive	Inactive
Primary	1-a-8-0/eth	Inactive	Inactive
Primary	1-b-8-0/eth	Inactive	Inactive
Primary	1-a-9-0/eth	Inactive	Inactive
Primary	1-b-9-0/eth	Inactive	Inactive
Primary	1-a-10-0/eth	Inactive	Inactive
Primary	1-b-10-0/eth	Inactive	Inactive
Primary	1-a-11-0/eth	Inactive	Inactive
Primary	1-b-11-0/eth	Inactive	Inactive

- 2 Break the line redundancy function on the port you will use for the EAPS ring.

```
zSH> line-red remove 1-b-3-0/eth
```

This command may take several minutes to complete.

Do you want to continue? (yes or no) [no] **yes**

Interface 1-b-3-0/eth is no longer in a protection group.

Entering a **line-red show all** will not display *1-a-3-0/eth* or *1-b-3-0/eth* because line is now concurrent and not redundant.

Facility protection on the MXK

With the 2.2.x release of the MXK, all of the Ethernet ports on uplink cards are redundant with corresponding ports on paired uplink cards. Port 1 of the uplink card in slot *a* is paired with port 1 of the uplink card in slot *b*; port 2 of the uplink card in slot *a* with port 2 of the uplink card in slot *b*, and so on. These pairing of ports for redundant uplinks are called line redundant groups and provide facility protection on the MXK.

If a link on a port in Active state goes down, traffic is automatically sent to the corresponding port on the Standby uplink. Previous versions of the MXK required the Active uplink card to switch to the Standby uplink card when a link went down, now just the downed link changes.

Uplink cards still function as Active and Standby. The Active uplink card maintains the system databases. In default state, the active ports of the line redundant groups are on the uplink card in slot *a*.

Since both uplink cards are in the RUNNING state, when an active link in a group goes down, the standby link in the group takes over and the state of both cards remains the same — RUNNING. Facility protection on redundant uplinks allows both the Active and the Standby card to pass traffic.

Redundant uplink configuration

Equipment protection

For equipment protection, the uplink cards are configured as Active and Standby. Traffic is handled by links on the active card. When the Active card fails, the Standby card takes over control of the system and all traffic is switched to the Standby card. See [MXK redundant uplinks for equipment protection configuration on page 626](#) for equipment protection configuration.

Single uplink card facility protection

Facility protection accomplished by using two or more network facing ports in a redundant configuration. If the primary link goes down, the secondary link will begin to carry traffic. See [Ethernet redundancy on page 1154](#) and [Create Ethernet line redundancy on page 1155](#) for instructions on how to configure line redundancy.

Facility protection

Facility protection across cards allows traffic to pass on either the Active or the Standby uplink card. The Active card still controls the system even when the Standby card is also running traffic.

For example, when the **slots** command is entered and RUNNING+TRAFFIC displays on both uplink card *a* and uplink card *b*, facility protection has been invoked and traffic is running across both uplink cards. The asterisk next to uplink card *a* indicates that uplink card *a* is the Active card.

```

zSH> slots
MXK 823
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
Cards
10: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM
(RUNNING)
12: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM
(RUNNING)

```

Configure card redundancy with the line-red command

When the uplink cards in slot *a* and slot *b* are redundant, you can configure a port on the uplink card to switch from the active card to the standby card when a link fails. This is recommended for any critical link that goes off the device. You use the **line-red set** command to configure a port for switchover.

Configuring an uplink port for redundancy

When a port is configured for link redundancy with the **line-red set** command and a link on the port fails, the active card reboots and all traffic on the active card is switched to the standby card. The standby card then changes state to active.

- 1 Verify any existing redundancy with the **line-red show all** command.

```

zSH> line-red show all
No line redundancy interfaces for system

```

- 2 Create redundancy on a port, in this case port 3.

```

zSH> line-red set 1-a-3-0/eth timeout 0

```

If a link fails on port 3, a failover will now be triggered.

- 3 Verify the redundancy.

```

zSH> line-red show 1-a-3-0/eth
redundancy status for 1-a-3-0/eth:
      REBOOT standbytx ENABLE timeout 0 NONREVERTIVE revert timeout 0
Interface-Type      Interface-Name      Oper-State Oper-Status
=====
Primary             1-a-3-0/eth             Active     UP
Secondary           1-b-3-0/eth             Standby    Trfc-Disable

```

Prepare an uplink port for EAPS configuration

All redundant uplink ports on the MXK for the 2.2 release start in a line-red state and only those links on the Active port will pass traffic.

For the ports on the MXK uplinks that pass traffic to the network, the line-red ability should remain.

For the ports on the MXK uplinks that are a part of an EAPS ring, the line-red needs to be broken so that the Standby uplink can pass traffic.

Removing line redundancy on the uplink card

To run the MXK uplink cards as concurrent, remove any previously existing line redundancies.

1 Verify the existence of line redundancies.

```
zSH> line-red show all
```

Line Redundant interfaces for system

Interface-Type	Interface-Name	Oper-State	Oper-Status
=====	=====	=====	=====
Primary	1-a-1-0/eth	Active	UP
Primary	1-b-1-0/eth	Standby	Trfc-Disable
Primary	1-a-2-0/eth	Inactive	Inactive
Primary	1-b-2-0/eth	Inactive	Inactive
Primary	1-a-4-0/eth	Inactive	Inactive
Primary	1-b-4-0/eth	Inactive	Inactive
Primary	1-a-5-0/eth	Active	UP
Primary	1-b-5-0/eth	Standby	Trfc-Disabl
Primary	1-a-6-0/eth	Inactive	Inactive
Primary	1-b-6-0/eth	Inactive	Inactive
Primary	1-a-7-0/eth	Inactive	Inactive
Primary	1-b-7-0/eth	Inactive	Inactive
Primary	1-a-8-0/eth	Inactive	Inactive
Primary	1-b-8-0/eth	Inactive	Inactive
Primary	1-a-9-0/eth	Inactive	Inactive
Primary	1-b-9-0/eth	Inactive	Inactive
Primary	1-a-10-0/eth	Inactive	Inactive
Primary	1-b-10-0/eth	Inactive	Inactive
Primary	1-a-11-0/eth	Inactive	Inactive
Primary	1-b-11-0/eth	Inactive	Inactive

2 Remove the line-red redundancy on the standby port to break the protection group.

```
zSH> line-red remove 1-b-5-0/eth
```

This command may take several minutes to complete.

Do you want to continue? (yes or no) [no] **yes**

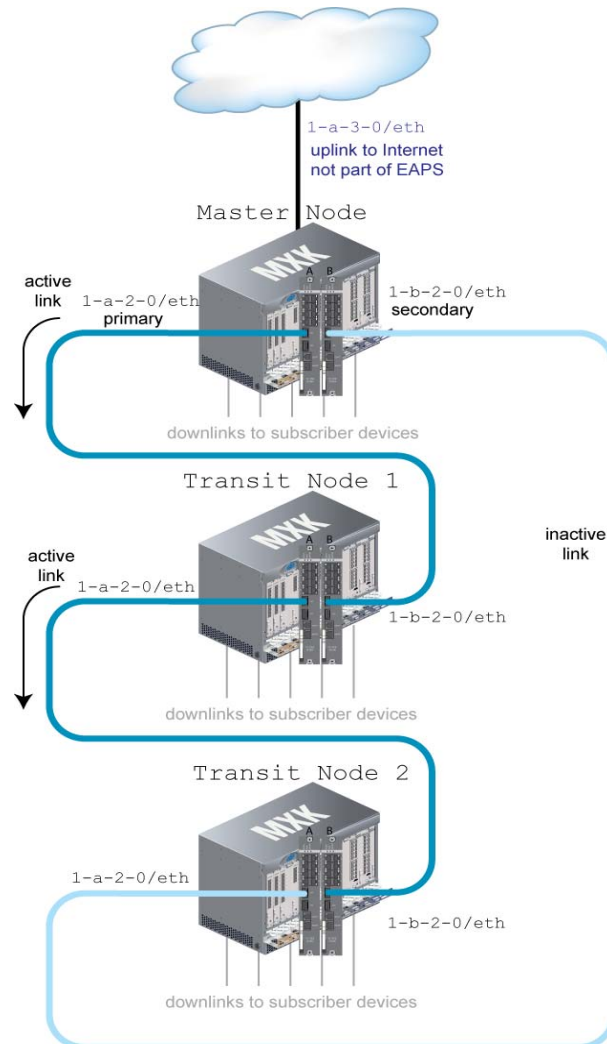
Interface 1-b-5-0/eth is no longer in a protection group.

EAPS

The Ethernet Automatic Protection Switching (EAPS) protocol (RFC 3619) creates a fault tolerant system of links by providing fast switchover from a primary path to a secondary path for each VLAN. This system is used to create a ring so that if one path is broken, the other direction may be used to get the packet to the destination.

In many access and Metro Area Network (MAN) topologies, transport between nodes, and from nodes to the distribution layer of the network is required to run over a physical ring infrastructure. In these cases, multi-platform support for a packet ring protocol is needed to connect and communicate over the ring topology.

Figure 89: An EAPS ring has a Master node and one or more transit nodes. Arrows show downstream direction



As described in RFC 3619, "An EAPS Domain exists on a single Ethernet ring. Any Ethernet Virtual Local Area Network (VLAN) that is to be protected is configured on all ports in the ring for the given EAPS Domain.

Each EAPS domain has a single ‘master node.’ All other nodes on that ring are referred to as ‘transit nodes.’ ”

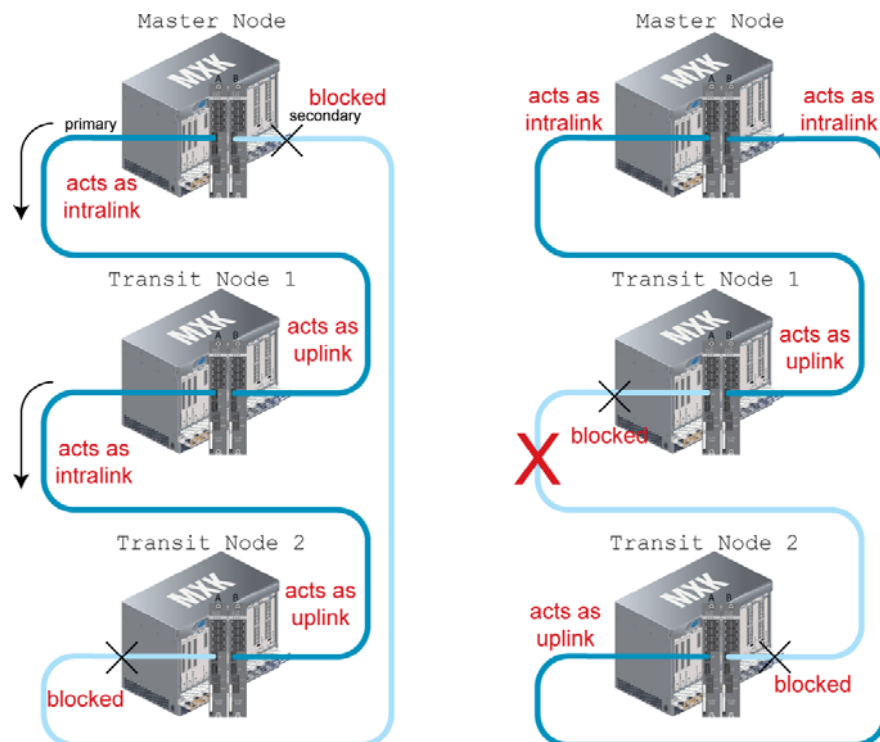
EAPS, which is only for layer 2 bridging, takes advantage of multiple VLANs being on the same physical link. One VLAN is a control VLAN which sends Health Check messages around the ring. Because a bridge cannot loop back on itself, there is always a blocked link for data on the EAPS ring, but not for the control VLAN, so the control messages can loop around, but data does not.

Figure 90: With multiple VLANs on a link one is a control VLAN which protects the other VLAN



Each master node on the ring has two ports which are part of the ring. One is the primary port and one secondary. EAPS supports both TLS and asymmetric bridges.

Figure 91: When a Link Down condition is detected the active interface and blocked interface change



For TLS the primary link on a master node is active and the secondary link is blocked. For TLS bridges the last link on the transit node which loops back to the master node (in the initial state) will be blocked, otherwise the links will be active.

For asymmetric bridges the primary port on the master node (in its initial state) will act as an intralink, the other port will be blocked. For the transit nodes one rlink (the name for these bridge interfaces on the transit node) will act as an uplink. If the other node is going to another transit node, the other rlink will act as an intralink. On the last link which loops back to the master node, the rlink will be blocked.



Note: EAPS is a flexible fault tolerance mechanism. The Zhone MXK product line is also very flexible. For questions regarding limitations and best practices when deploying EAPS solutions, contact your Zhone Sales Engineer.

If a transit node detects that a link has gone down, it sends a Link Fault message to the master. So any time there is a delayed Health Check message or a Link Fault message the master will change the direction traffic goes to the other nodes.

When a link is down EAPS messages on the control VLAN notify the other nodes which essentially clears their bridging data bases and switches how each link will act. For a master node, the blocked node will function as an intralink. The other link on the master node and rlinks adjust as well depending upon on which link the break occurs.

Recommendations for success using EAPS

The master node to the ring is safest if it has no subscriber services and uses asymmetric (uplink and intralink) bridge interfaces.

The asymmetric model is best for creating larger and safer EAPS rings. With uplink and intralink on the master node and rlinks on the transit node, the traffic is moved around the ring via VLAN without requiring multicasts to determine which interface to send the traffic down, or learning MAC addresses. When TLS is used both the multicasts and the MAC learning will occur, which will put an extra, and usually, unnecessary load on the system.

If TLS must be used it is best if it is isolated to a node. That is, if you have multiple TLS bridges working together that do not require going across one of the rlinks. Once you cross rlinks then you have the issue of multicast and learning MAC addresses that can multiply very quickly. At that point the options of what subscriber services are on the nodes can get complex in terms of load in combination with EAPS very quickly, so you should seek assistance from your network consultant and Zhone representative in designing your EAPS network.

Creating asymmetric and TLS EAPS rings

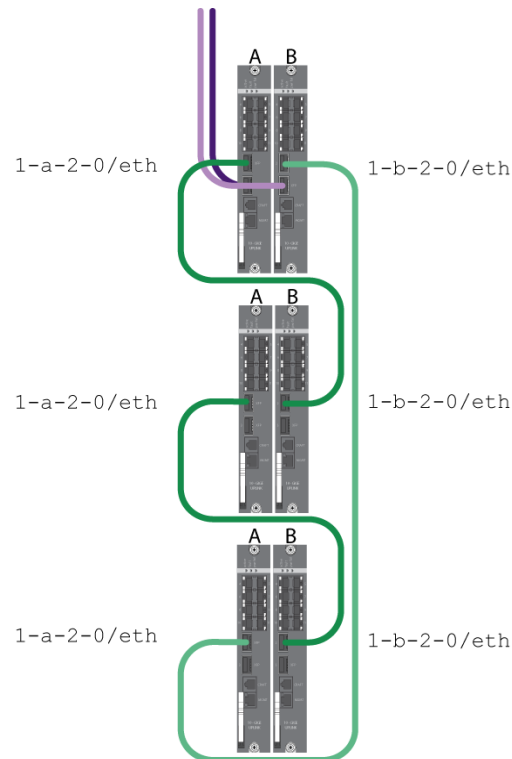
EAPS may be used with asymmetric and TLS bridges. In fact, since the control VLAN is just another VLAN, asymmetric and TLS bridges could possibly be on the same physical interfaces.



Note: Asymmetric bridges are recommended. See [Recommendations for success using EAPS, page 642](#) for more information. For options managing EAPS rings, see [Managing inband using IP on a bridge with EAPS, page 657](#).

Asymmetric EAPS

Figure 92: Asymmetric EAPS example



To create an EAPS ring (asymmetric example)

The following steps create an EAPS ring as shown in [Figure 92](#). Since upstream and downstream directions are important with asymmetric bridges, the term **rlink** is used to designate the bridge interfaces on the transit nodes.

- 1 Configure the master node
 - a Break the uplink line redundancy

Since uplink ports default to a line redundant state when pairs of uplinks are used the line redundant needs to be removed for the pair. To remove the line redundancy you need to remove one port from the group, the other line will not have a second, so the redundancy is removed from that port as well.

```
zSH> line-red remove 1-b-2-0/eth
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
Interface 1-b-2-0/eth is no longer in a protection group.
```

See [Configure line-red uplink ports for concurrent EAPS \(2.2.x\)](#), page 635 for more information.

- b** Configure the node as a master node, define the **domain_id**, define the primary and secondary interfaces and define the control VLAN

```
zSH> eaps add domain EAPSAsymExample master
1-a-2-0/eth 1-b-2-0/eth control 100
```

If you do not state values for interval, timeout and ctrlpri, the defaults will be used:

- **interval**, 1 second
- **timeout**, 3 seconds
- **ctrlpri** (control VLAN priority), 6

See [EAPS commands on page 664](#) for more detail.

- c** Add the protected VLANs to the domain

```
zSH> eaps-vlan add domain EAPSAsymExample
200-300,999
```

- d** Enable the node

```
zSH> eaps enable domain EAPSAsymExample
```

- e** Add the bridges to the node

```
zSH> bridge add 1-a-3-0/eth uplink vlan 200 tagged
(upstream to cloud, not part of eaps)
zSH> bridge add 1-a-2-0/eth intralink vlan 200
tagged (primary link)
zSH> bridge add 1-b-2-0/eth intralink vlan 200
tagged (secondary link)
```

Intralinks are recommended instead of downlinks because downlinks will save the MAC addresses in the forwarding database.



Note: Since the master node is the connection upstream to the rest of the network, it is best not to put a heavy throughput load on the subscriber access ports.

2 Configure the first transit node**a** Break the uplink line redundancy

Once again following the Zhone standard with the EAPS ring on **1-a-2-0/eth** and **1-b-2-0/eth**, we will break the line redundancy.

```
zSH> line-red remove 1-b-2-0/eth
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
Interface 1-b-2-0/eth is no longer in a protection group.
```

See [Configure line-red uplink ports for concurrent EAPS \(2.2.x\)](#), page 635 for more information.

b Configure the node as a transit node, define the **domain_id**, define the primary and secondary interfaces and define the control VLAN

```
zSH> eaps add domain EAPSAsymExample transit
1-a-2-0/eth 1-b-2-0/eth control 100
```

c Add the protected VLANs to the domain

```
zSH> eaps-vlan add domain EAPSAsymExample
200-300,999
```

d Enable the node

```
zSH> eaps enable domain EAPSAsymExample
```

e Add the bridges to the node

```
zSH> bridge add 1-a-2-0/eth rlink vlan 200 tagged
zSH> bridge add 1-b-2-0/eth rlink vlan 200 tagged
```

3 Configure the second transit node**a** Break the uplink line redundancy

Again following the Zhone standard with the EAPS ring on **1-a-2-0/eth** and **1-b-2-0/eth** we will break the line redundancy.

```
zSH> line-red remove 1-b-2-0/eth
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
Interface 1-b-2-0/eth is no longer in a protection group.
```

See [Configure line-red uplink ports for concurrent EAPS \(2.2.x\)](#), page 635 for more information.

b Configure the node as a transit node, define the **domain_id**, define the primary and secondary interfaces and define the control VLAN

```
zSH> eaps add domain TransitNode2 transit 1-a-2-0/
eth 1-b-2-0/eth control 100
```

c Add the protected VLANs to the domain

```
zSH> eaps-vlan add domain EAPSAsymExample
200-300,999
```

- d** Enable the node

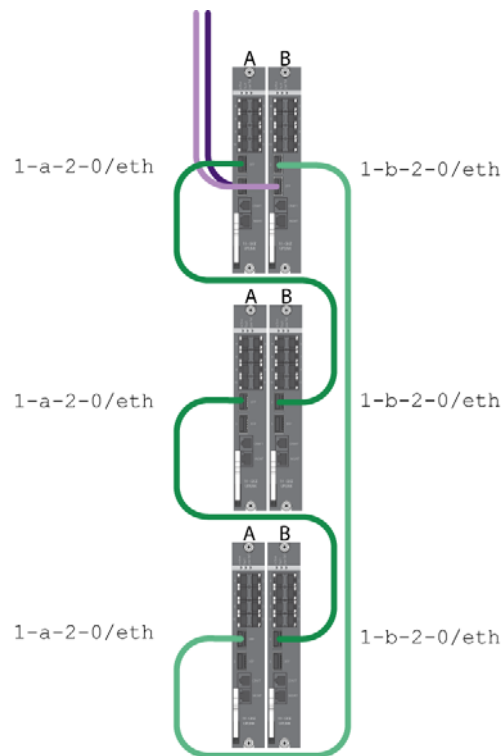
```
zSH> eaps enable domain EAPSAsymExample
```

- e** Add the bridges to the node

```
zSH> bridge add 1-a-2-0/eth rlink vlan 200 tagged
zSH> bridge add 1-b-2-0/eth rlink vlan 200 tagged
```

TLS EAPS

Figure 93: TLS EAPS example



To create an EAPs ring (TLS example)

The only difference between creating TLS interfaces with EAPS rather than with asymmetric bridges is that instead of designating the bridge interface as an **rlink** as you would for asymmetric bridges, the bridge interface is designated as **tls**. The TLS example is shown in [Figure 93](#).

- 1** Configure the master node

- a** Break the uplink line redundancy

Since uplink ports default to a line redundant state when pairs of uplinks are used the line redundant needs to be removed for the pair. To remove the line redundancy you need to remove one port from the group, the other line will not have a second, so the redundancy is removed from that port as well.

```
zSH> line-red remove 1-b-2-0/eth
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
Interface 1-b-2-0/eth is no longer in a protection group.
```

See [Configure line-red uplink ports for concurrent EAPS \(2.2.x\)](#), page 635 for more information.

- b** Configure the node as a master node, define the **domain_id**, define the primary and secondary interfaces and define the control VLAN

```
zSH> eaps add domain MasterNode_tls master 1-a-2-0/
eth 1-b-2-0/eth control 102
```

If you do not state values for interval, timeout and ctrlpri, the defaults will be used:

- interval, 1 second
- timeout, 3 seconds
- ctrlpri (control VLAN priority), 6

See [EAPS commands on page 664](#) for more detail.

- c** Add the protected VLANs to the domain

```
zSH> eaps-vlan add domain EAPSTLSExample
400-499,998
```

- d** Enable the node

```
zSH> eaps enable domain EAPSTLSExample
```

- e** Add the bridges to the node

```
zSH> bridge add 1-a-3-0/eth tls vlan 400 tagged
(upstream to cloud, not part of eaps)
zSH> bridge add 1-a-2-0/eth tls vlan 400 tagged
(primary link)
zSH> bridge add 1-b-2-0/eth tls vlan 400 tagged
(secondary link)
```

2 Configure the first transit node

- a** Break the uplink line redundancy

Once again following the Zhone standard with the EAPS ring on **1-a-2-0/eth** and **1-b-2-0/eth**, we will break the line redundancy.

```
zSH> line-red remove 1-b-2-0/eth
This command may take several minutes to complete.
```

Do you want to continue? (yes or no) [no] **yes**
 Interface 1-b-2-0/eth is no longer in a protection group.

See [Configure line-red uplink ports for concurrent EAPS \(2.2.x\), page 635](#) for more information.

- b** Configure the node as a transit node, define the **domain_id**, define the primary and secondary interfaces and define the control VLAN

```
zSH> eaps add domain EAPSTLSEExample transit
1-a-2-0/eth 1-b-2-0/eth control 102
```

- c** Add the protected VLANs to the domain

```
zSH> eaps-vlan add domain EAPSTLSEExample
400-499,998
```

- d** Enable the node

```
zSH> eaps enable domain EAPSTLSEExample
```

- e** Add the bridges to the node

```
zSH> bridge add 1-a-2-0/eth tls vlan 400 tagged
zSH> bridge add 1-b-2-0/eth tls vlan 400 tagged
```

3 Configure the second transit node

- a** Break the uplink line redundancy

Again following the Zhone standard with the EAPS ring on **1-a-2-0/eth** and **1-b-2-0/eth** we will break the line redundancy.

```
zSH> line-red remove 1-b-2-0/eth
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
Interface 1-b-2-0/eth is no longer in a protection group.
```

See [Configure line-red uplink ports for concurrent EAPS \(2.2.x\), page 635](#) for more information.

- b** Configure the node as a transit node, define the **domain_id**, define the primary and secondary interfaces and define the control VLAN

```
zSH> eaps add domain EAPSTLSEExample transit
1-a-2-0/eth 1-b-2-0/eth control 102
```

- c** Add the protected VLANs to the domain

```
zSH> eaps-vlan add domain EAPSTLSEExample
400-499,998
```

- d** Enable the node

```
zSH> eaps enable domain EAPSTLSEExample
```

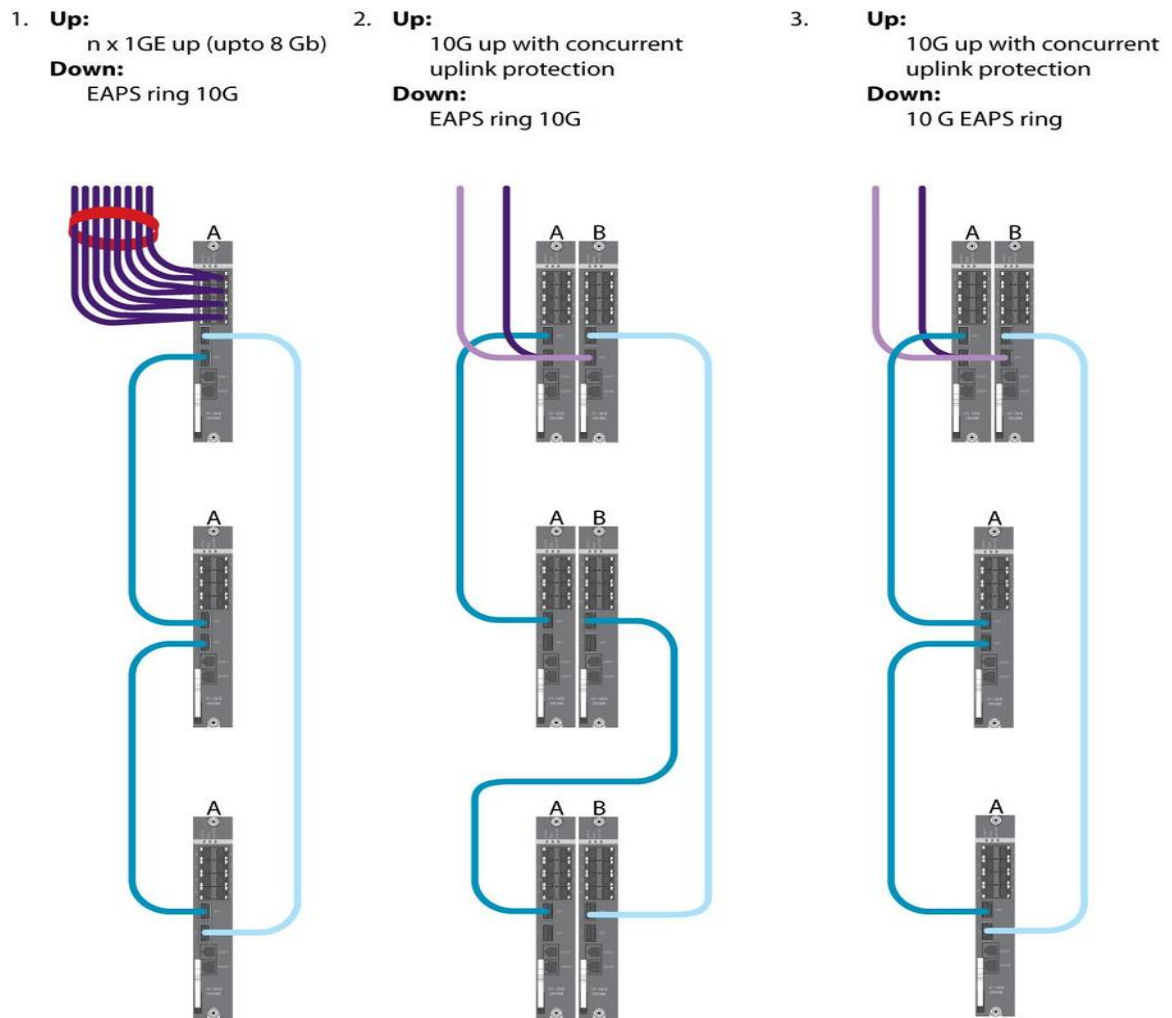
- e** Add the bridges to the node

```
zSH> bridge add 1-a-2-0/eth t1s vlan 400 tagged
zSH> bridge add 1-b-2-0/eth t1s vlan 400 tagged
```

Common EAPs topologies

EAPS rings may be created on either 10G or 1GE links. Some nodes may have dual uplinks (usually as the Master node), other nodes may have a single uplink.

Figure 94: The most common EAPS scenarios may combine dual and single uplinks



Note: Zhone does not support EAPS rings over link aggregation groups.

EAPS topology command

To see the topology of an EAPS ring from the CLI, use the **eaps topo** or the **eaps topo2** command. The **eaps show** command will display the overall status of the ring and control VLAN information.

```
zSH> eaps show
Primary      Secondary   Domain      Status      Type
CtrlVlan
-----
example_simple_ringActive-HealthyM1-a-10-01-b-11-04089
```

The eaps topo command shows which node you are viewing the topology from (both by the ***** and the 0 hops), the MAC address of the uplink card, the IP address (if there is one), whether the node is a master node or a transit node (**M/T**). If a link is down an “x” will after the shelf-slot-port-subport designation will show the location of the break in the ring.

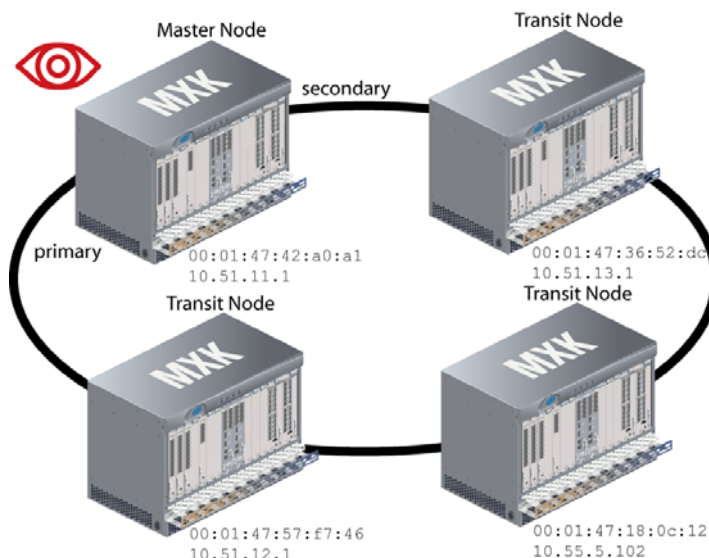
The older eaps topo command which is still available via the **eaps topo2** command shows which node you are viewing the topology from (both by the ***** and the 0 hops), the MAC address of the uplink card, the IP address (if there is one), whether the node is a master node or a transit node (**M/T**) and the interface and status of the link (**up/dn**). If a link is down the shelf-slot-port-subport designation will have dn after it. The eaps topo2 command also shows the path from the viewing node around both ways, so that there will be 2n-1 lines for the number of nodes on the ring. So for a healthy ring with four nodes there will be seven lines as shown in the eaps topo2 example. For an eaps ring with a break there will be entries for each node that can be reached.

eaps topo

The eye in the diagram shows on which node the command is run; notice that the display confirms the node with the *****. The command shows the EAPS nodes each way until it reaches the node which it was run from, or there is a break in the ring.

The header information shows the domain name as well as the control VLAN and the state of the EAPS ring.

Figure 95: A healthy four node ring



zSH> eaps topo

Domain "4nodes" [CtrlVlan:4089,State:Active-Healthy]:

Ring	Hop	Mac Addr	IP Addr	Type	Links
-----*****-----					
10.51.30.1	M	1-b-2-0	00:01:47:6d:55:56		
					1-a-2-0
Pri	1	00:01:47:6d:55:42	10.51.32.1	T	1-b-2-0
					1-a-2-0
Pri	2	00:01:47:36:52:db	10.55.5.105	T	1-b-2-0
					1-a-2-0
Pri	3	00:01:47:57:f7:3e	10.55.5.104	T	1-b-2-0
					1-a-2-0

If there is a break in the ring the break is designated by an “x” where the interfaces detect the break.

zSH> eaps topo

Domain "4nodes" [CtrlVlan:4089,State:Ring-Fault]:

Ring	Hop	Mac Addr	IP Addr	Type	Links

```
-----
-----***** 0 00:01:47:6d:55:56
10.51.30.1      M 1-b-2-0 x

                                1-a-2-0

Pri    1  00:01:47:6d:55:42      10.51.32.1  T 1-b-2-0

                                1-a-2-0

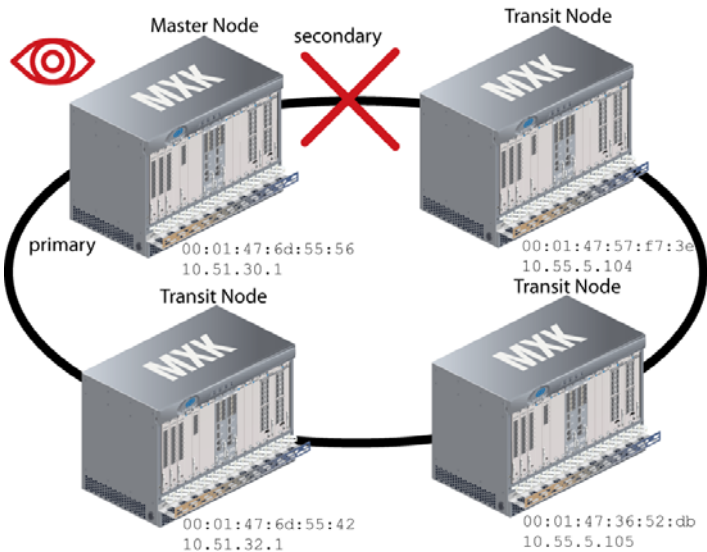
Pri    2  00:01:47:36:52:db      10.55.5.105 T 1-b-2-0

                                1-a-2-0

Pri    3  00:01:47:57:f7:3e      10.55.5.104 T 1-b-2-0

                                1-a-2-0 x
```

Figure 96: The topo command will display the break



The above example has the break on the unused secondary link from the master however the break is still reported.

Understanding how the interfaces from different nodes connect together can be quite important when troubleshooting. With the **eaps topo** command you can follow which interface connects to which interface easily. The 1-a-20 interface on the master connects to the 1-b-2-0 interface on the first transit node. The first transit node interface to the second transit node is 1-a-2-0 on the first and 1-b-2-0 on the second. The unused secondary (in a healthy ring) from the last MXK in the ring is 1-a-2-0 to the secondary interface on the master 1-b-2-0. If we were to break the ring on that link you would see the “x” marking the broken link.

Figure 97: Pattern of interface connections with eaps topo display

4port-Master> eaps topo2

Domain "4port" [CtrlVlan:4089,State:Ring Fault]:

Ring	Hop	Mac Addr	IP Addr	Type	Links

Pri	3	00:01:47:57:f7:3e	10.55.5.104	T	1-b-2-0(up) 1-a-2-0(up)
Pri	2	00:01:47:36:52:db	10.55.5.105	T	1-b-2-0(up) 1-a-2-0(up)
Pri	1	00:01:47:6d:55:42	10.51.32.1	T	1-b-2-0(up) 1-a-2-0(up)
****	0	00:01:47:6d:55:56	10.51.30.1	M	1-a-2-0(up) 1-b-2-0(dn)
Sec	1	00:01:47:6d:55:54	10.51.31.1	T	1-b-2-0(up) 1-a-2-0(up)
Sec	3	00:01:47:57:f7:3e	10.55.5.104	T	1-b-2-0(up) 1-a-2-0(up)

With system logging on, alerts will be shown in the monitoring CLI session.

```
FEB 26 17:36:53: alert : 1/a/0 : eapsrp: Domain
ZhoneRtlEaps: Preforwarding --> Ring Fault
FEB 26 17:39:28: alert : 1/a/0 : eapsrp: Domain
ZhoneRtlEaps: Ring Fault --> Active-Healthy
FEB 26 18:44:21: alert : 1/a/1033: clitask3: User
admin@10.57.100.100 logged in on slot a
```

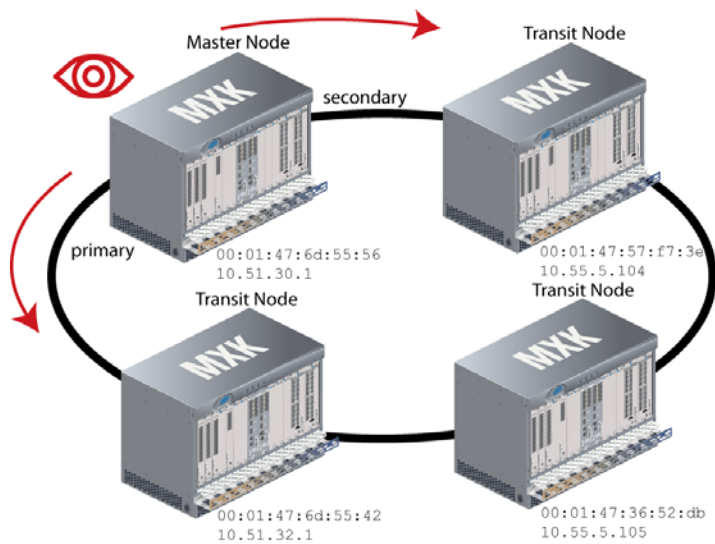
```
FEB 26 18:55:57: alert : 1/a/1027: clitask0: User admin
logged in on slot a
```

If a link is down, the **eaps topo** command shows the link down status with the x designation.

eaps topo2

Like the **eaps topo** command the eye in the diagram for the **eaps topo2** command shows on which node the command is run; notice that the display confirms the node with the ********. The command shows the EAPS nodes each way until it reaches the node which it was run from, or there is a break in the ring.

Figure 98: A healthy four node ring



The header information shows the domain name as well as the control VLAN and the state of the EAPS ring.

```
zSH> eaps topo
```

```
Domain "4nodes" [CtrlVlan:4089,State:Active-Healthy]:
```

Ring	Hop	Mac	Addr	IP	Addr	Type	Links

----- Pri			3	00:01:47:57:f7:3e			
10.55.5.104	T	1-b-2-0 (up)	1-a-2-0 (up)				

Pri	2	00:01:47:36:52:db		10.55.5.105		T	
1-b-2-0 (up)	1-a-2-0 (up)						

Pri	1	00:01:47:6d:55:42		10.51.32.1		T	
1-b-2-0 (up)	1-a-2-0 (up)						

****	0	00:01:47:6d:55:56		10.51.30.1		M	
1-a-2-0 (up)	1-b-2-0 (up)						

Sec	1	00:01:47:6d:55:54		10.51.31.1		T	
1-b-2-0 (up)	1-a-2-0 (up)						

Sec	2	00:01:47:8b:d7:2e		10.51.33.1		T	
1-b-2-0 (up)	1-a-2-0 (up)						

Sec	3	00:01:47:57:f7:3e		10.55.5.104		T	
1-b-2-0 (up)	1-a-2-0 (up)						

Unlike the **eaps topo** command, with the **eaps topo2** command it is a little harder to read which interfaces on one node connect to interfaces on another node. From the ******** in the display you read upward, the 1-a-2-0 on the master node walks up to the first interface (1-b-2-0) on transit node 1. Then it

walks up the display in a regular pattern. A similar pattern follows for reading down the display.

Figure 99: Pattern of interface connections with eaps topo2 display

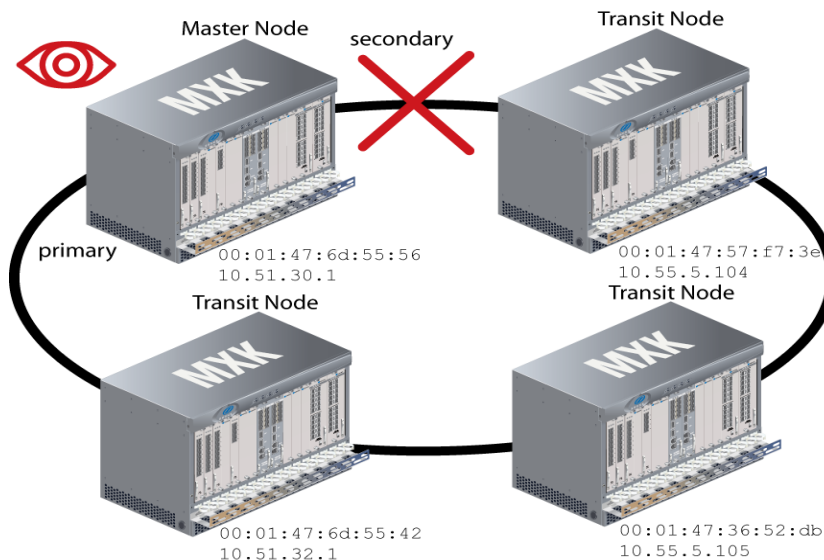
```
4port-Master> eaps topo2
```

Domain "4port" [CtrlVlan:4089,State:Ring Fault]:					
Ring Hop	Mac Addr	IP Addr	Type	Links	

Pri 3	00:01:47:57:f7:3e	10.55.5.104	T	1-b-2-0(up) 1-a-2-0(up)	
Pri 2	00:01:47:36:52:db	10.55.5.105	T	1-b-2-0(up) 1-a-2-0(up)	
Pri 1	00:01:47:6d:55:42	10.51.32.1	T	1-b-2-0(up) 1-a-2-0(up)	
**** 0	00:01:47:6d:55:56	10.51.30.1	M	1-a-2-0(up) 1-b-2-0(up)	
Sec 1	00:01:47:6d:55:54	10.51.31.1	T	1-b-2-0(up) 1-a-2-0(up)	
Sec 2	00:01:47:8b:d7:2e	10.51.33.1	T	1-b-2-0(up) 1-a-2-0(up)	
Sec 3	00:01:47:57:f7:3e	10.55.5.104	T	1-b-2-0(up) 1-a-2-0(up)	

If there is a break in the ring the break is designated by the (dn) where the interfaces detect the break.

Figure 100: The topo command will display until the break



```
zSH> eaps topo
```

-----Domain "4port"					
[CtrlVlan:4089,State:Ring Fault]:					
Ring Hop	Mac Addr	IP Addr	Type	Links	

Pri 3	00:01:47:57:f7:3e	10.55.5.104	T	1-b-2-0(up) 1-a-2-0(dn)	
Pri 2	00:01:47:36:52:db	10.55.5.105	T	1-b-2-0(up) 1-a-2-0(up)	
Pri 1	00:01:47:6d:55:42	10.51.32.1	T	1-b-2-0(up) 1-a-2-0(up)	

```

****      0  00:01:47:6d:55:56      10.51.30.1      M
1-a-2-0(up) 1-b-2-0(dn)
Sec      1  00:01:47:6d:55:54      10.51.31.1      T
1-b-2-0(up) 1-a-2-0(up)
Sec      3  00:01:47:57:f7:3e      10.55.5.104      T
1-b-2-0(up) 1-a-2-0(up)

```

With system logging on, alerts will be shown in the monitoring CLI session.

```

FEB 26 17:36:53: alert   : 1/a/0      : eapsrp: Domain
ZhoneRtlEaps: Preforwarding --> Ring Fault
FEB 26 17:39:28: alert   : 1/a/0      : eapsrp: Domain
ZhoneRtlEaps: Ring Fault --> Active-Healthy
FEB 26 18:44:21: alert   : 1/a/1033: clitask3: User
admin@10.57.100.100 logged in on slot a

FEB 26 18:55:57: alert   : 1/a/1027: clitask0: User admin
logged in on slot a

```

If a link is down, the **eaps topo2** command shows the link down status with the (dn) designation.

Configure line-red state for concurrent EAPS ports (2.2.x and later)

With the MXK release 2.2.x, all uplink ports default to a **line-red** state when uplink cards reside in slots *a* and *b* and share the same line group number. When a link down occurs on a port, traffic switches from the Active port to the Standby port. This redundancy support does not allow an EAPS ring to use the paired ports of a pair of uplink cards to pass traffic. EAPS requires both ports to pass traffic, both for the control VLAN and the active or blocked ports.



Note: On the physical uplink card, the green light will blink when traffic has been switched from an active to a standby port. In the EAPS case, traffic will be running on uplink cards in both slots *a* and *b*. If an uplink card needs to be pulled, traffic will be switched to one card. EAPS will deal with the card removal like a line down condition.

Changing an uplink port from line-red to concurrent for EAPS ring configuration

For concurrent uplinks that will be part of an EAPS ring, the **line-red** state on the uplink port must be broken so that traffic can be passed on ports on both cards, the standby as well as the active card. Breaking the **line-red** link on concurrent uplink cards allows for the control VLAN (heartbeat) pass traffic and allows for EAPS to control the traffic behavior of the links.

- 1 Verify the state of the uplink ports with the **line-red show all** command.

```
zSH> line-red show all
```

```
Line Redundant interfaces for system
```

Interface-Type	Interface-Name	Oper-State	Oper-Status
Primary	1-a-1-0/eth	Active	UP
Primary	1-b-1-0/eth	Standby	Trfc-Disable
Primary	1-a-2-0/eth	Inactive	Inactive
Primary	1-b-2-0/eth	Inactive	Inactive
Primary	1-a-3-0/eth	Inactive	Inactive
Primary	1-b-3-0/eth	Inactive	Inactive
Primary	1-a-4-0/eth	Inactive	Inactive
Primary	1-b-4-0/eth	Inactive	Inactive
Primary	1-a-5-0/eth	Active	UP
Primary	1-b-5-0/eth	Standby	OOS
Primary	1-a-6-0/eth	Inactive	Inactive
Primary	1-b-6-0/eth	Inactive	Inactive
Primary	1-a-7-0/eth	Inactive	Inactive
Primary	1-b-7-0/eth	Inactive	Inactive
Primary	1-a-8-0/eth	Inactive	Inactive
Primary	1-b-8-0/eth	Inactive	Inactive
Primary	1-a-9-0/eth	Inactive	Inactive
Primary	1-b-9-0/eth	Inactive	Inactive
Primary	1-a-10-0/eth	Inactive	Inactive
Primary	1-b-10-0/eth	Inactive	Inactive
Primary	1-a-11-0/eth	Inactive	Inactive
Primary	1-b-11-0/eth	Inactive	Inactive

- 2 Break the line redundancy function on the port you will use for the EAPS ring.

```
zSH> line-red remove 1-b-3-0/eth
```

```
This command may take several minutes to complete.
```

```
Do you want to continue? (yes or no) [no] yes
```

```
Interface 1-b-3-0/eth is no longer in a protection group.
```

Entering a **line-red show all** will not display *1-a-3-0/eth* or *1-b-3-0/eth* because line redundancy no longer functions on these ports.

Managing inband using IP on a bridge with EAPS

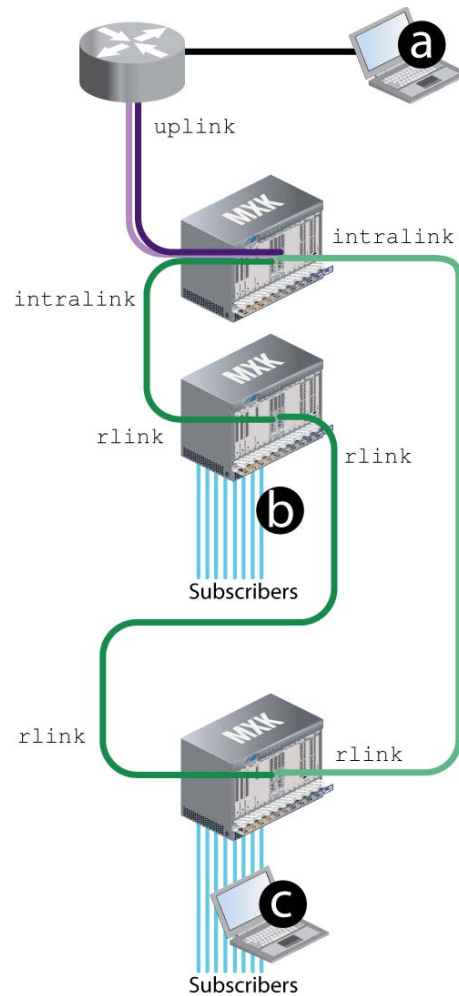
EAPS is a bridged solution. For management via IP, an ipobridge interface is created on the node as a management interface.

Management on an asymmetric EAPS ring

The master node should be set up with an uplink to the network and intralinks; the transit nodes use rlinks as described in [Asymmetric EAPS on page 643](#).

On the master node an ipobridge should be created on a management VLAN on one of the intralinks. For the transit nodes an ipobridge should be created on the management VLAN on one of the rlinks.

Figure 101: To manage inband from within the network the master EAPS node needs to be a separate subnet



Since EAPS is a layer 2 bridging solution, the router will learn the MAC addresses for the top connector EAPS master, but with intralinks will not learn the transit nodes unless they are in a different subnet.

If all the nodes are in the same IP subnet, you will only be able to manage the nodes in the ring from outside the ring as designated in the graphic by PC a.

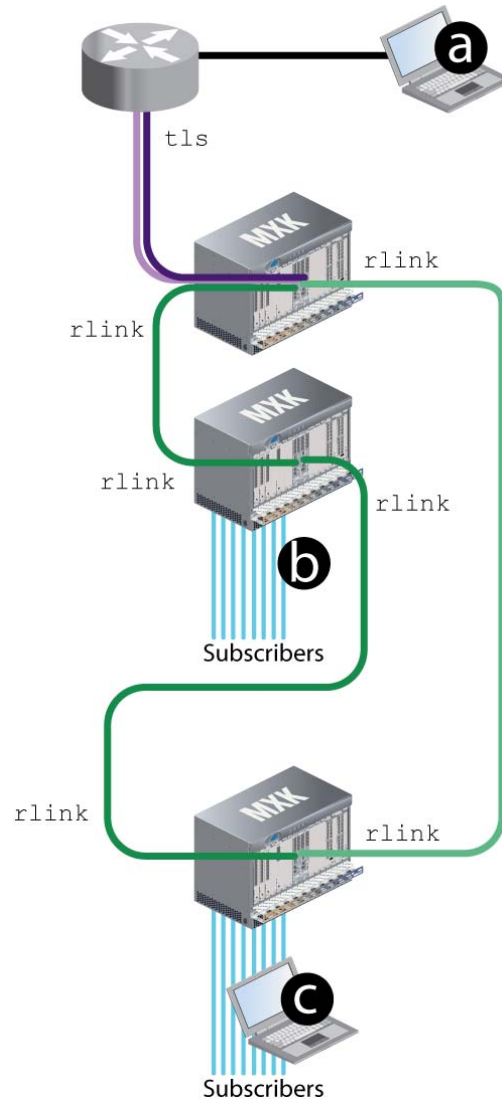
To manage from within the asymmetric EAPS ring, the ipobridge interface for the master node should be in a different IP subnet than the transit nodes. The transit nodes may be in the same IP subnet. It is okay for the transit nodes to be in the same IP subnet. That way management of the EAPS ring may be done from the PCs designated b and c in the graphic.

Management on a TLS EAPS ring

For building a TLS EAPS ring see [TLS EAPS on page 646](#). Configuring management on a TLS EAPS ring is simpler than on an asymmetric ring. For an EAPS ring you put the ipobridge on one of the rlinks on each node, whether master or transit node.

Since with TLS it will do multicast to find MAC addresses and store MAC addresses when found all three possible locations for a management device are supported even when the entire EAPS ring is in the same IP subnet.

Figure 102: TLS EAPS example



Creating IP on a bridge interfaces for an EAPS ring

The following commands assume a master node and two transit nodes. Along with other VLANs, VLANs 300, 301 and 302 are added as protected VLANs. As in the configuration examples described above, the uplink uses linkagg ([To create an EAPs ring \(asymmetric example\), page 643](#) and [To create an EAPs ring \(TLS example\), page 646](#)).

Notice that while the physical or virtual interface is given, the ipobridge is built upon the VLAN, so only one ipobridge interface need be created per TLS node. The ipobridge will be TLS if built on a VLAN that is TLS and will be a downlink if built on a VLAN that is asymmetric.

1 Add the ip interface on the master node

```
zSH> interface add 1-a-1-0/linkagg vlan 300
192.168.120.1/24
```

The master node should be set up with an uplink to the network and intralinks.

2 Add the ipobridge interface on one transit node

```
zSH> interface add 1-a-6-0/ipobridge vlan 301
193.168.121.1/24
```

the transit nodes must use TLS bridge interfaces because the **interface add ipobridge** command automatically creates a TLS bridge interface on the VLAN.



Note: Asymmetric bridge interfaces (uplink, downlink, intralink, rlink) may not be used on the same VLAN as TLS interfaces.

3 Add the ipobridge interface on the other transit node

```
zSH> interface add 1-a-6-0/ipobridge vlan 302
194.168.122.1/24
```

the transit nodes must use TLS bridge interfaces because the **interface add ipobridge** command automatically creates a TLS bridge interface on the VLAN.

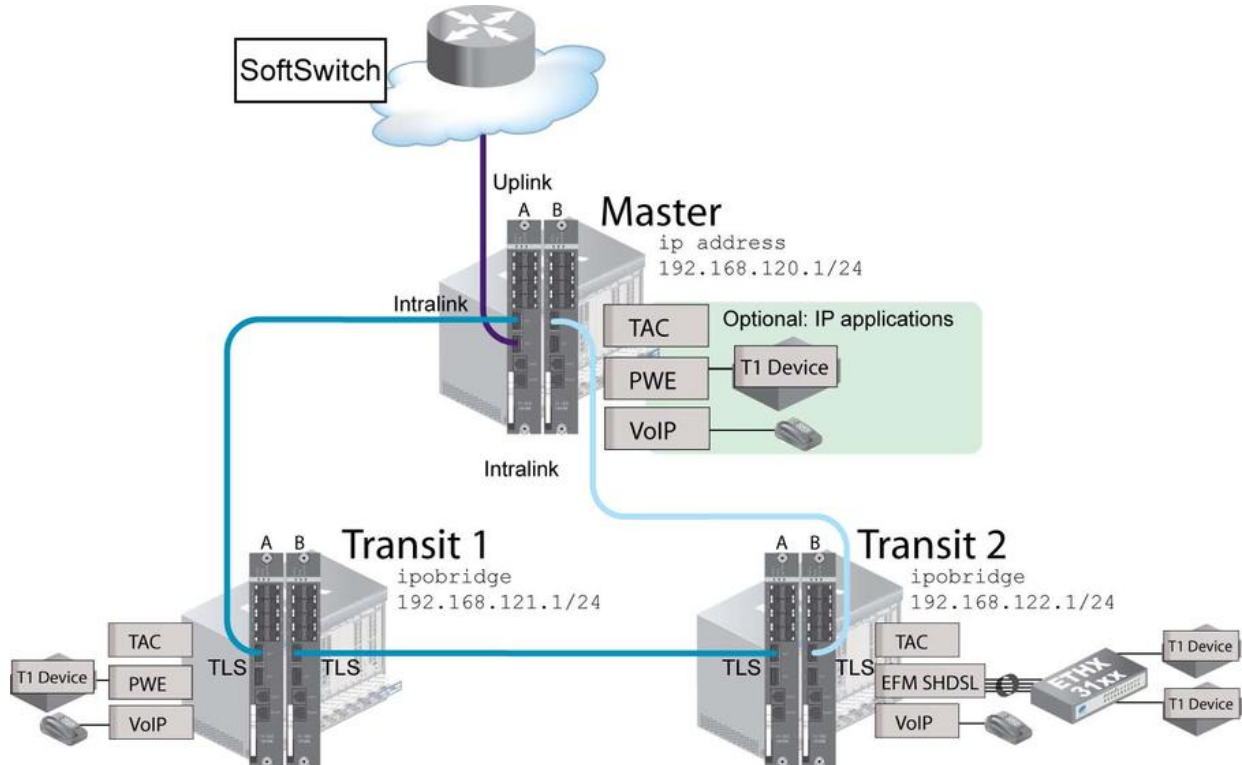


Note: Asymmetric bridge interfaces (uplink, downlink, intralink, rlink) may not be used on the same VLAN as TLS interfaces.

IP applications using IP on a bridge with EAPS

EAPS is a bridged solution. Some applications such as VoIP and PWE are IP based solutions. To combine EAPS with these solutions an ipobridge interface is created on the node as a management interface.

Figure 103: When adding an ipobridge to a transit node, the other bridge interfaces must be TLS bridge interfaces



Configuring IP on a bridge with EAPS

The master node should be set up with an uplink to the network and intralinks; the transit nodes must use TLS bridge interfaces because the **interface add ipobridge** command automatically creates a TLS bridge interface on the VLAN. Asymmetric bridge interfaces (uplink, downlink, intralink, rlink) may not be used on the same VLAN as TLS interfaces.

The VoIP application will use the IP address of the softswitch as normal, but the source MXK needs to provide an IP address via ipobridge.

For PWE devices accessing far end PWE devices in the cloud, the far end target is addressed as normal. The end shown here (the target for the remote end uses the IP address from the ipobridge interface.

For PWE devices accessing other PWE which are attached to devices in the EAPS ring, the IP addresses of the ipobridge interface are use. If the PWE solution on Transit 1 needs to access the PWE solution on Transit 2, the packet stream will go to the router in the cloud, then back through the EAPs ring to the proper IP address.

Each transit node uses a single VLAN for the IP applications, whether the application is VoIP or PWE. It is recommended to use a separate VLAN for each transit node, as that will improve switch over performance if a link fails.

1 Configure the master node

a Break the uplink line redundancy

```
zSH> line-red remove 1-b-2-0/eth
```

This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] **yes**
Interface 1-b-2-0/eth is no longer in a protection group.

b Configure the node as a master node, define the **domain_id**, define the primary and secondary interfaces and define the control VLAN

```
zSH> eaps add domain EAPSPWEEexample master 1-a-2-0/  
eth 1-b-2-0/eth control 4089
```

c Add the protected VLANs to the domain

```
zSH> eaps-vlan add domain EAPSPWEEexample  
200-399,600-699
```

d Enable the node

```
zSH> eaps enable domain EAPSPWEEexample
```

e Add the bridges to the node

```
zSH> bridge add 1-a-3-0/eth uplink vlan 300 tagged  
(up to cloud, IP apps on master)  
zSH> bridge add 1-a-3-0/eth uplink vlan 301 tagged  
(up to cloud, IP apps on trans1)  
zSH> bridge add 1-a-3-0/eth uplink vlan 302 tagged  
(up to cloud, IP apps on trans2)
```

```
zSH> bridge add 1-a-2-0/eth intralink vlan 301  
tagged (primary link, IP apps on trans1)  
zSH> bridge add 1-a-2-0/eth intralink vlan 302  
tagged (primary link, IP apps on trans2)
```

```
zSH> bridge add 1-b-2-0/eth intralink vlan 301  
tagged (secondary link, IP apps on trans1)
```

```
zSH> bridge add 1-b-2-0/eth intralink vlan 302  
tagged (secondary link, IP apps on trans2)
```

f Add the ip interface on the uplink

```
zSH> interface add 1-a-3-0/ip vlan 300  
192.168.120.1/24
```

2 Configure the first transit node**a** Break the uplink line redundancy

Once again following the Zhone standard with the EAPS ring on **1-a-2-0/eth** and **1-b-2-0/eth**, we will break the line redundancy.

```
zSH> line-red remove 1-b-2-0/eth
```

This command may take several minutes to complete.
 Do you want to continue? (yes or no) [no] **yes**
 Interface 1-b-2-0/eth is no longer in a protection group.

See [Configure line-red uplink ports for concurrent EAPS \(2.2.x\)](#), page 635 for more information.

- b** Configure the node as a transit node, define the **domain_id**, define the primary and secondary interfaces and define the control VLAN

```
zSH> eaps add domain EAPSPWEEexample transit
1-a-2-0/eth 1-b-2-0/eth control 4089
```

- c** Add the protected VLANs to the domain

```
zSH> eaps-vlan add domain EAPSPWEEexample
200-399,600-699
```

- d** Enable the node

```
zSH> eaps enable domain EAPSPWEEexample
```

- e** Add the bridges to the node

```
zSH> bridge add 1-a-2-0/eth tls vlan 301 tagged
(primary link, IP apps for trans1)
zSH> bridge add 1-b-2-0/eth tls vlan 301 tagged
(secondary link, IP apps for trans1)
```

- f** Add the ipobridge interface

```
zSH> interface add 1-a-6-0/ipobridge vlan 301
192.168.121.1/24
```

Notice that this is a different subnet since this is a different VLAN

3 Configure the second transit node

- a** Break the uplink line redundancy

Once again following the Zhone standard with the EAPS ring on **1-a-2-0/eth** and **1-b-2-0/eth**, we will break the line redundancy.

```
zSH> line-red remove 1-b-2-0/eth
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
Interface 1-b-2-0/eth is no longer in a protection group.
```

- b** Configure the node as a transit node, define the **domain_id**, define the primary and secondary interfaces and define the control VLAN

```
zSH> eaps add domain EAPSPWEEexample transit
1-a-2-0/eth 1-b-2-0/eth control 4089
```

- c** Add the protected VLANs to the domain

```
zSH> eaps-vlan add domain EAPSPWEEExample
200-399,600-699
```

- d** Enable the node

```
zSH> eaps enable domain EAPSPWEEExample
```

- e** Add the bridges to the node

```
zSH> bridge add 1-a-2-0/eth t1s vlan 302 tagged
(primary link, IP apps for trans1)
```

```
zSH> bridge add 1-b-2-0/eth t1s vlan 302 tagged
(secondary link, IP apps for trans1)
```

- f** Add the ipobridge interface

```
zSH> interface add 1-a-6-0/ipobridge vlan 302
192.168.122.1/24
```

Notice that this is a different subnet since this is a different VLAN

EAPS commands

eaps

The **eaps** command configures an EAPS master node or transit node.

Syntax `eaps <add|delete|disable|enable|help|modify|show|topo>`

Syntax `eaps add domain domain_id <master|transit> control vlan <interface1> interface1/type <interface2> interface2/type <interval value> <drop value> <timeout value> <trap [yes|no]> <cntlpri value>`

The **eaps add** command creates a node as an EAPS master or transit node, defines which interfaces are primary and secondary interfaces for the node, defines the control VLAN for the EAPS node (the control VLAN must be the same as the other nodes in the EAPS ring), and defines other variables such as the time between Health Check messages, the amount of time before determining a link fault, whether to send SNMP alert messages and defining the priority of messages on the control and protected VLANs. A newly added node is placed in the Inactive state

domain domain_id

domain is the alpha-numerical value (i.e. a string name) of the domain. It is user assigned and should represent the string description of the Domain's purpose (for example, which ring the node belongs to as in "South_Campus", which node, "transit_node_2", or a combination "South_Campus_transit_node_2").

master|transit

The **master|transit** parameter defines the state of the node on a ring; only one node for a given domain and control VLAN can be master, all other nodes are required to be transit nodes.

<interface1> interface1/type

The **interface1** parameter defines the first port; if a node is configured as Master, then **interface1** will be considered the primary port. The string token "interface1" is optional on the **eaps add** command, but mandatory for the **eaps modify** command

<interface2> interface2/type

The **interface2** parameter defines the first port; if a node is configured as Master, then **interface2** will be considered the primary port. The string token "interface2" is optional on the **eaps add** command, but mandatory for the **eaps modify** command

control vlan

The **control** parameter denotes the control VLAN (outermost VLAN for Q-in-Q non-802.1Q support) for the EAPS ring Domain. All nodes on the same EAPS ring must use the same control VLAN.

interval time

The interval parameter sets the time in seconds between Health messages (frequency) to be transmitted out of the primary ring port. The interval parameter is only valid for master nodes. The default value used when **interval** is not explicitly stated in the command is one second.

timeout time

The **timeout** parameter sets the timeout duration in seconds. For a Master Node this is the time since the last receipt of a HELLO message before the master node declares a ring fault. The default value used when **timeout** is not explicitly stated in the master form of the command is three seconds.

For a Transit Node this is the time the Node will remain in state Preforwarding before it transitions (in the event of a dropped a Ring Up message) to the Health state. The default value used when **timeout** is not explicitly stated in the transit form of the command is fifteen seconds.

trap <on | off>

The trap parameter determines whether the node will send an SNMP alert describing the state change.

cntlpri priority

The **cntlpri** parameter sets the priority that Control messages will be sent. The default value used when **cntlpri** is not explicitly stated in the transit form of the command is six.

Syntax `eaps modify domain domain_id <master|transit> <control vlan> <interface1 interface1/type> <interface2 interface2/type> <interval value> <drop value> <timeout value> <trap [yes|no]> <cntlpri value>`

The **eaps modify** command changes the configured variables of and EAPS node. The EAPS node must be disabled before using the **eaps modify** command.

Syntax `eaps show domain <domain_id|all>`

The **eaps show** command displays the protected VLAN list for the domain if the list exists. The command returns an error if the supplied **domain_id** does not exist.

domain domain_id|all>

domain is the alpha-numerical value (i.e. a string name) of the domain. It is user assigned and should represent the string description of the Domain's purpose (for example, which ring the node belongs to as in "South_Campus", which node, "transit_node_2", or a combination "South_Campus_transit_node_2").

all displays all eaps connections on the node.

Syntax `eaps topo domain domain_name`

Display topology information regarding the EAPS rings of which this node is a participating node.

Syntax `eaps disable domain domain_id`

The **eaps disable** command disables an existing domain on the node. (The node no longer moves traffic. To modify a node using the **eaps modify** command, the node must first be disabled.)

Syntax `eaps enable domain domain_id`

The **eaps enable** command enables an existing domain on the node. The **eaps enable** command moves the domain from a disabled state to an enabled state, so the node may pass messages on protected VLANs.

eaps-vlan

The **eaps-vlan** command creates, modifies, deletes or displays protected VLANs on an EAPS node.

Syntax

Syntax `eaps-vlan <add | delete | modify | show > domain domain_id VLAN_List`

Syntax `eaps-vlan add domain domain_id VLAN_List`

The **eaps-vlan add** command creates a Protected VLAN or a list of protected VLANs for an existing domain. The command will return an error if the supplied domain_id does not already exist.

Protected VLAN lists consist of comma separated entries, an entry can be either a single VLAN Id, or a dash separated range of ascending VLAN Ids. VLAN Ids must be greater than zero and less than 4090. You may use at most four entries. No spaces are permitted. e.g. "eaps-vlan add domain metro2 100-113,205,370-420,999" would create a list of protected VLANs for Domain "metro2" consisting of VLANs 100 through 113, VLAN 205, VLANs 370 through 420 and VLAN 999.

Syntax `eaps-vlan delete domain domain_id VLAN_List`

The **eaps-vlan delete** command removes a Protected VLAN or a list of protected VLANs for an existing domain. The command will return an error if the supplied *domain_id* does not already exist.

Syntax `eaps-vlan modify domain domain_id VLAN_List`

The **eaps-vlan modify** command alters a Protected VLAN or a list of protected VLANs for an existing domain. The command will return an error if the supplied *domain_id* does not already exist.

Syntax `eaps-vlan show domain <domain_id|all>`

The **eaps-vlan modify** command displays the list of protected VLANs for an existing domain. The command will return an error if the supplied *domain_id* does not already exist.

Displaying and updating Ethernet interfaces

The **list**, **get**, and **update** commands support use of the interface *shelf-slot-port-subport/eth* syntax to facilitate Ethernet port and interface monitoring and configuration.

To list the currently configured Ethernet interfaces, enter the **list ether** command.

```
zSH> list ether
ether 1-a-1-0/eth
ether 1-a-2-0/eth
ether 1-a-3-0/eth
ether 1-a-4-0/eth
ether 1-a-5-0/eth
ether 1-a-6-0/eth
ether 1-a-7-0/eth
ether 1-a-8-0/eth
ether 1-a-9-0/eth
ether 1-a-10-0/eth
ether 1-a-11-0/eth
ether 1-b-1-0/eth
ether 1-b-2-0/eth
ether 1-b-3-0/eth
ether 1-b-4-0/eth
ether 1-b-5-0/eth
ether 1-b-6-0/eth
ether 1-b-7-0/eth
ether 1-b-8-0/eth
ether 1-b-9-0/eth
ether 1-b-10-0/eth
ether 1-b-11-0/eth
ether 1-13-1-0/eth
ether 1-13-2-0/eth
ether 1-13-3-0/eth
ether 1-13-4-0/eth
...
42 entries found.
```

The **list ether** command shows the Ethernet interfaces on each uplink card, in slot a and slot b, as well as the Ethernet interfaces on the Active Ethernet card.

The **slots** command verifies the location of the cards with Ethernet interfaces:

```
zSH> slots
MXK 819
Uplinks
  a: *MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
  4: MXK 4 PORT GPON (RUNNING)
  13: MXK 20 ACT ETH (RUNNING)
```


To view an Ethernet interface, enter the **get ether** command followed by the interface index.

```
zSH> get ether 1-a-4-0/eth
ether 1-a-4-0/eth
autonegstatus: ----> {enabled}
mauType: -----> {mau1000baselxfd}
restart: -----> {norestart}
ifType: -----> {mau1000baselxfd}
autonegcap: -----> {b10baseTFD+b100baseTXFD+b1000baseTFD}
remotefault: -----> {noerror}
clksrc: -----> {automatic}
pauseFlowControl: -> {disabled}
aggregationMode: --> {on}
linkStateMirror: --> {0/0/0/0/0}
```

Small form factor pluggables

Zhone Technologies supports a variety of small form factor pluggables (SFPs) and XFPs which are selected depending on the protocol, fiber type and distance requirements. For information and specifications on supported SFPs and XFPs, see [Chapter 18, Small Form Factor Pluggable \(SFP\) Connectors](#), on page 1117.

Uplink card pinouts

This section lists the pinouts for the following interfaces that are common on all the uplink cards:

- [Serial \(craft\) port pinouts](#), page 670
- [Ethernet port pinouts](#), page 671

For information about other port pinouts for uplink cards, refer to the chapters for each type of card, later in this manual.

Serial (craft) port pinouts

[Table 60](#) lists the uplink cards’ serial (craft) port pinouts. The serial (craft) port is an RS232 D type configured as DTE.

Table 60: Uplink card serial (craft) port pinouts

Pin	Function
1	DCE Ready, Ring Indicator (DSR/RI)
2	Received Line Signal Detector (DCD)
3	DTE Ready (DTR)
4	Signal Ground (SGND)
5	Received Data (RD)
6	Transmitted Data (TD)
7	Clear To Send (CTS)
8	Request To Send (RTS)

[Table 61](#) lists the pinouts to connect a DB9 connector to the MXK RJ45 serial craft port.

Table 61: RJ45 to DB9 adapter pinouts

RJ-45 pin	Color	Function	DB-9 pin
1	N/A	DCE Ready, Ring Indicator (DSR/RI)	not used
2	N/A	Received Line Signal Detector (DCD)	not used
3	N/A	DTE Ready (DTR)	not used
4	Red	Signal Ground (SGND)	5
5	Green	Received Data (RD)	2
6	Yellow	Transmitted Data (TD)	3
7	N/A	Clear To Send (CTS)	Looped to pin 8
8	N/A	Request To Send (RTS)	Looped to pin 7

Ethernet port pinouts

[Table 62](#) lists the Ethernet port pinouts on the uplink cards.

Table 62: Uplink card Ethernet port pinouts

Pin	Function
1	Tx +
2	Tx -
3	Rx +
4	Not used
5	Not used
6	Rx -
7	Not used
8	Not used

10

MXK GPON CARDS

This chapter describes the MXK Gigabit Passive Optical Networks (GPON) cards and GPON card configuration:

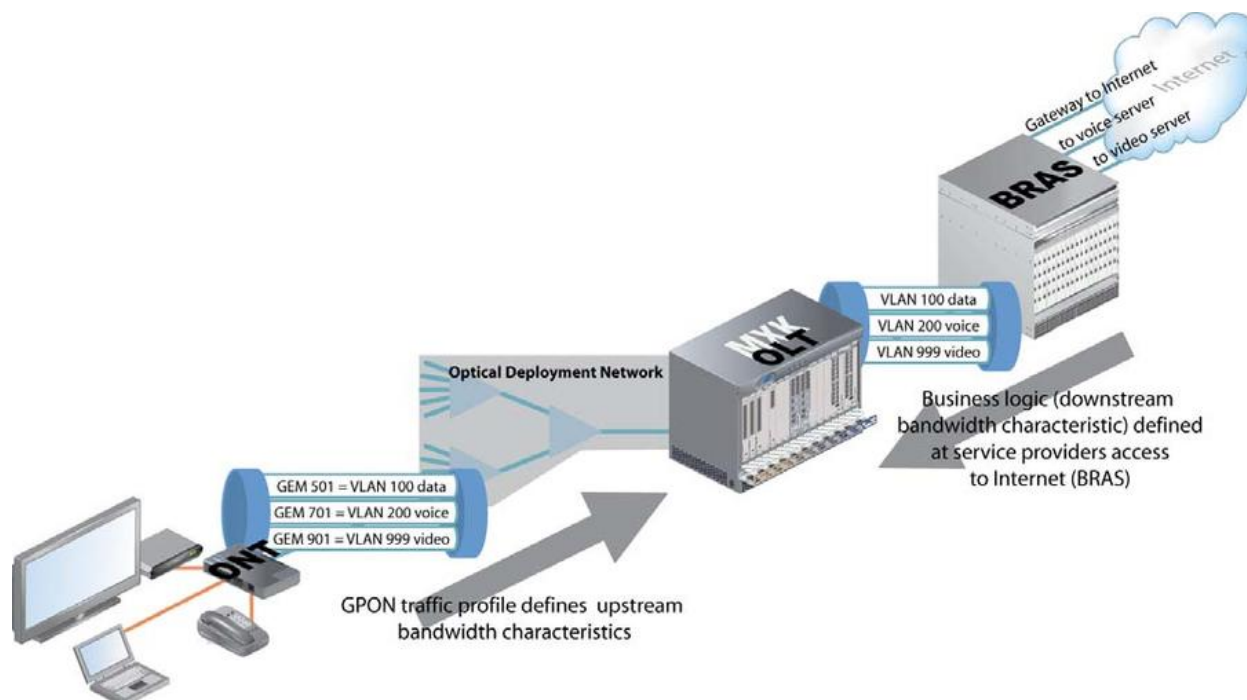
- [GPON cards, page 675](#)
- [GPON on the MXK, page 679](#)
- [Smart OMCI GPON zNID installation, page 692](#)
- [Unified Service Provisioning GPON zNID installation, page 725](#)
- [ONU Software Upgrades, page 900](#)
- [Manage ONU with OMCI, page 910](#)
- [MXK GPON using the Reg ID for provisioning, page 923](#)
- [Bandwidth Allocation for Upstream Traffic from the ONU to the MXK, page 925](#)
- [GEM port creation, page 938](#)
- [GPON ONU serial number format \(Hexadecimal or Decimal\), page 945](#)
- [Received Signal Strength Indication \(RSSI\) and Digital Diagnostic Monitoring \(DDM\), page 948](#)
- [Configurable range for Reserved VLAN per GEM port, page 951](#)
- [GPON type B redundancy, page 956](#)
- [GPON extended reach, page 965](#)
- [GPON Business Applications, page 968](#)
- [ONT Inventory Report, page 970](#)
- [OMCI Statistics, page 972](#)
- [PON Statistics, page 975](#)
- [GPON Alarms and Traps, page 987](#)

MXK GPON provides a ITU-T G.984 GPON standards-based fiber-based GPON solution.

GPON provides a maximum of 2.5 Gbps downstream and 1.25 Gbps upstream traffic. GPON is a point-to-multipoint architecture which may be split up to 64 subscriber ends, so the 2.5 Gbps downstream/1.25 Gbps

upstream is split among the subscribers. All information is sent out to all units. Encryption keeps information private.

Figure 104: Where the MXK and the Optical Deployment Network fits in the GPON solution.



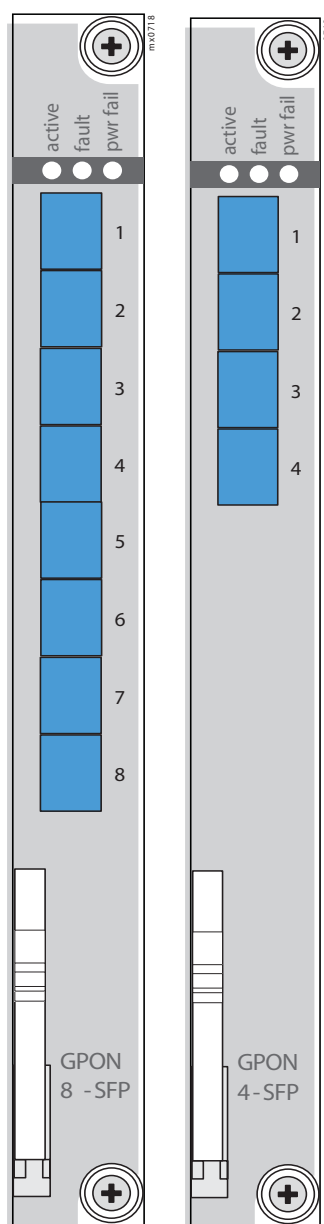
As shown in [Figure 104](#), business logic such as downstream bandwidth is defined at the service providers access to the Internet. Upstream bandwidth characteristics are defined in the GPON traffic profiles.

GPON cards

This section describes the MXK GPON cards and how to configure the cards.

- [GPON card overview, page 675](#)
- [GPON card specifications, page 676](#)
- [GPON card configuration, page 677](#)
- [View additional card and system information, page 678](#)

GPON card overview



Zhone provides two GPON line cards, the MXK-GPONX8-IO and the MXK-GPONX4-IO. Both GPON cards support 2.5 Gbps downstream bandwidth and 1.25 Gbps upstream bandwidth per interface as specified in the G.984.1-4 specifications.

The MXK-GPONX8-IO line card has an octal-port interface that provides industry leading capabilities. The MXK 8 port GPON card can support up to 512 GPON subscribers.

The MXK-GPONX4-IO line card has a quad-port interface. The MXK 4 port GPON card can support up to 256 GPON subscriber.

The SFPs used in the MXK GPON cards are:

- MXK-GPON-SFP-B+-RSSI
- MXK-GPON-SFP-C+-RSSI

These two SFPs support Received Signal Strength Indication (RSSI) feature. For more information about RSSI, please see [Received Signal Strength Indication \(RSSI\) and Digital Diagnostic Monitoring \(DDM\)](#) on page 948.

AES encryption of 128 bits is supported on the GPON OLT chipset.

See [Chapter 5, IP Configuration](#), [Chapter 4, MXK Bridge Configuration](#), and [Chapter 5, Video Configuration](#) for procedures on configuring routes, bridges, and video.

The following features are supported:

- Class C+ Optics with -32dB link budget, 60 km maximum reach
- Class B+ Optics with -28dB link budget, 20 km reach
- 64 subscribers per OLT interface
- RF Overlay (1550nm wavelength)
- GPON type B redundancy
- Traffic management for IP QoS, traffic shaping
- RSSI support

GPON card specifications

[Table 63](#) provides the MXK-GPONX8-IO and the MXK-GPONX4-IO card specifications.

Table 63: GPON OLT card specifications

Specification	Value
Size	1 slot
Density	8 port – 512 subscribers (64 subscribers per interface) 4 port – 256 subscribers (64 subscribers per interface)
Physical interfaces	SC-UPC fiber optic connector.
Line characteristics	Receives wavelength at a 1310nm Transmits wavelength at a 1490nm
Nominal line rate	2.5 Gbps downstream 1.25 Gbps upstream
Protocol support	Multicast IGMP v2 Network-based routing IP host and gateway support DHCP server (RFC 2131, 2132), DHCP relay Bridging 802.1D VLAN 802.1Q/p Dense/sparse multicast
Power consumption	63 W (8 port) 50 W (4 port)

GPON card configuration

Each card installed in the system must have a **card-profile**. The line card type determines the parameter settings in the **card-profile** and the software image for the card. Performing a **card add <slot #>** automatically creates the **card-profile** for the card with the correct software image and settings.

[Table 64](#) provides the type and software image for the GPON cards on the MXK.

Table 64: MXK GPON card types

Card	Type	Name of software image
MXK-GPONX8-IO	10203	<i>mxlc8gp.bin</i>
MXK-GPONX4-IO	10205	<i>mxlc4gp.bin</i>

Creating a card-profile for a GPON card

Each card installed in the system must have a **card-profile**. The **card-profile** is created when users perform a **card add**.

- 1 Install a GPON card in the desired line card slot.
- 2 Configure a GPON card on the MXK:

```
zSH> card add 5
```

The system creates a **card-profile** for the GPON card in *slot 5*.

Verifying the line card installation

- 1 After adding a card to the MXK, users can verify the card by entering **slots**:

```
zSH> slots
Uplinks
  a:*MXK EIGHT GIGE (RUNNING)
  b: MXK EIGHT GIGE (RUNNING)
Cards
  1: MXK 8 PORT GPON (RUNNING)
  5: MXK 4 PORT GPON (RUNNING)
```

- 2 Enter the **slots** command and specify the slot number of the card to view card information and the state of the card. For example:

```
zSH> slots 5
Type           : MXK 4 PORT GPON
Card Version    : 800-02586-01-A
EEPROM Version  : 1
Serial #       : 1963593
CLEI Code      : No CLEI
Card-Profile ID : 1/5/10205
```

```

Shelf          : 1
Slot           : 5
ROM Version    : MXK 1.15.1.108
Software Version: MXK 1.16.2.119
State          : RUNNING
Mode           : FUNCTIONAL
Heartbeat check : enabled
Longest hbeat  : 79
Fault reset    : enabled
Uptime         : 32 days, 19 hours, 14 minutes

```

3 View the **card-profile** of the card by entering **get card-profile shelf/slot/cardtype**:

```

zSH> get card-profile 1/5/10205
card-profile 1/5/10205
sw-file-name: -----> {mxlc4gp.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {unknowntype}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}

```

View additional card and system information

View the EPROM version of the card:

```

zSH> eeshow card 5
EEPROM contents: for slot 5
EEPROM_ID    : 00 -- CARD
Version      : 01
Size         : 054
CardType     : 10205 -- MXLC4GP
CardVersion  : 800-02586-01-A
SerialNum    : 01963593
ShelfNumber  : 00001
CLEI Code    : No CLEI
Cksum        : 0x23E6

```

View the version of the system software:

```

zSH> swversion
Zhone mxUp2Tg8g software version MXK 1.16.2.119

```

GPON on the MXK

This section includes:

- [GPON terminology, page 679](#)
- [Bridge add commands with ranges of Slots, OLTs, GEM ports, and UNI ports, page 682](#)
- [Planning GPON networks, page 689](#)
- [Installation testing, page 690](#)
- [Handling fiber, page 691](#)

Although this guide is primarily concerned with configuring Zhone equipment, it is also important to have a strong understanding of the underlying technology. This section defines some general concepts to consider, and is not to be a definitive resource.



Note: All the commands that start with **gpononu** or **gponolt** can be replaced to start with **onu** or **olt**. For example: > **gpononu set** is same as > **onu set** ; > **gponolt show bw** is same as > **olt show bw**.

GPON terminology

This section describes:

- [Components of GPON optical deployment networks, page 679.](#)
- [Relationship between T-ports and GEM ports, page 680.](#)

Components of GPON optical deployment networks

Optical networks are comprised of a number of components between the subscriber devices.

- OLT
Optical Line Terminator. This device is considered the head end of the ODN. (Note that each port on a GPON line card is considered an OLT.)
- Optical fiber
The optical fiber is the physical cable.
- Optical splitters (GPON only)
Optical splitters split a single optical signal to multiple optical signals.
- Couplers
Couplers are connectorized means for splicing cables. Because couplers are connectors there is an optical signal cost for connectors

- ONT or ONU

Optical Network Terminator (ONT) and Optical Network Unit (ONU) are reasonably similar terms which are both defined in the ITU-T G.984 GPON standards. They both provide an end for the ODN and conversion to some electrical media; However, ONTs usually have multiple subscriber-side services and interfaces, like Ethernet LAN, POTS or coaxial cable for TV services. ONUs would have a GPON interface upstream (just like the ONT), but downstream direction they provide last mile copper access device such as a VDSL2 or Fast Ethernet which connects to customer premises equipment such as a VDSL2 modem or an Ethernet Hub, Switch or Residential Gateway.

- Attenuators

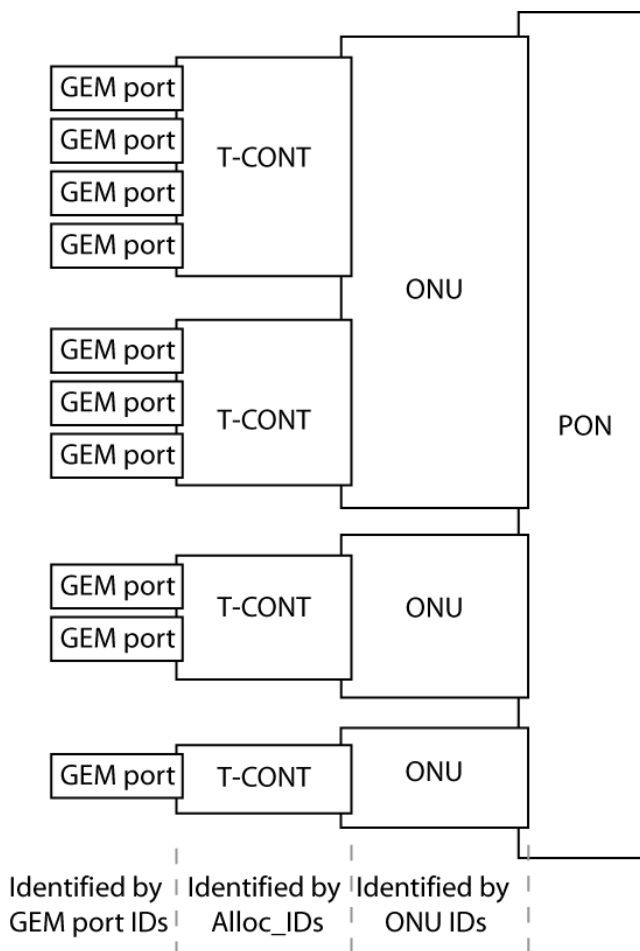
Attenuation is the term for the loss of optical power on the ODN. Some devices may actually receive too high a signal power strength for the receiving device. This situation most commonly occurs in lab settings. An attenuator can adjust the power strength of the optical signal.

All the fiber components named above are important in planning and installing GPON networks.

Relationship between T-ports and GEM ports

[Figure 105](#) shows the relationship between T-ports and GEM ports.

Figure 105: Relationship between T-contrs and GEM ports



- T-Contrs

Transmission Container (T-cont)s are how the ONU represents a group of logical connections that appear as a single traffic-bearing entity for the purpose of bandwidth assignment on the upstream side of the ONU.

Each ONU contains one or more T-contrs. The OLT discovers the number of T-contrs supported by a given ONU and assigns Alloc IDs to T-contrs in this ONU. Alloc ID is the identifier of a T-cont.

Each T-cont contains one or more GEM ports. The Alloc ID is assigned to a T-cont during the GEM port creation.

Bandwidth allocation on a T-cont is defined in the GPON traffic profile. Multiple GEM ports can share one T-cont by enabling shared feature in the associated GPON traffic profile.

- GEM ports

GPON Encapsulation Method (GEM) ports are how the ONUs separate the services from the upstream side of the ONU to the downstream ports. Each of these GEM ports needs to be unique on the ODN for the OLT port.

GEM port ID is the identifier of a GEM port. There are two types of GEM port IDs, Dynamic GEM port IDs used in Smart OMCI provisioning and Arbitrary GEM port IDs used in Unified Service Provisioning.

GEM ports are dynamically created during the **bridge add** operation. Conversely, GEM ports can be automatically deleted during the **bridge delete** operation. When creating a GEM port, a GPON traffic profile (that defines T-cont) must be used as well.

The traffic shaping on a GEM port is defined in a CPE traffic management Profile.

For detailed configurations and additional information on GEM ports, refer to:

[Dynamic GEM ports on page 695](#) (For Smart OMCI Provisioning)

[GEM Ports Assignments in USP on page 729](#) (For Unified Service Provisioning)

[GEM port creation on page 938](#)

Bridge add commands with ranges of Slots, OLTs, GEM ports, and UNI ports

In the **bridge add** command for GPON, the slotIDs, OLT port IDs, GEM port IDs, Ethernet UNI port IDs, and WLAN UNI port IDs may be replaced with brackets containing numbers in (comma-separated) series and/or (dash-separated) ranges. For Ethernet UNI ports and WLAN UNI ports, the wildcard “all” could be used too.

Here are some examples to specify port IDs in series, ranges and wildcards in the **bridge add** command for Smart OMCI and Unified Service Provisioning. Note that when specifying GEM ports in a range, Unified Service Provisioning must use the **bridge add** command with *shelfID-slotID-OLTportID-GEMportID/gponport* format.

- For Smart OMCI

- Slot IDs in a range

This example specifies the slot ID to 1 and 2, OLT port ID to 1, GEM port ID to 505, and ONU ID is implied to 5.

```
zSH> bridge add 1-[1,2]-1-505/gponport gtp 1 downlink vlan 505 tagged
To Abort the operation enter Ctrl-C
Adding bridge on 1-1-1-505/gponport
Created bridge-interface-record 1-1-1-505-gponport-505/bridge
Adding bridge on 1-2-1-505/gponport
Created bridge-interface-record 1-2-1-505-gponport-505/bridge
```

- OLT port IDs in a range

This example specifies the slot ID to 1, OLT port IDs to 2, 3, 4, and 6, GEM port ID to 501, and ONU ID is implied to 1.

```

zSH> bridge add 1-1-[2-4,6]-501/gponport gtp 1 downlink vlan 104 tagged
To Abort the operation enter Ctrl-C
Adding bridge on 1-1-2-501/gponport
Created bridge-interface-record 1-1-2-501-gponport-104/bridge
Adding bridge on 1-1-3-501/gponport
Created bridge-interface-record 1-1-3-501-gponport-104/bridge
Adding bridge on 1-1-4-501/gponport
Created bridge-interface-record 1-1-4-501-gponport-104/bridge
Adding bridge on 1-1-6-501/gponport
Created bridge-interface-record 1-1-6-501-gponport-104/bridge

```

- GEM port IDs in a range

This example specifies the slot ID to 1, OLT ID to 1, GEM port ID to 902 and 921, and ONU ID is implied to 2 and 21.

```

zSH> bridge add 1-1-1-[902,921]/gponport gtp 1 downlink vlan 3811 tagged
To Abort the operation enter Ctrl-C
Adding bridge on 1-1-1-902/gponport
Created bridge-interface-record 1-1-1-902-gponport-3811/bridge
Adding bridge on 1-1-1-921/gponport
Created bridge-interface-record 1-1-1-921-gponport-3811/bridge

```

- For Unified Service Provisioning

- Slot IDs in a range

This example specifies the slot ID to 1 and 2, OLT port ID to 1, ONU ID to 5, GEM port ID to 505.

```

zSH> bridge add 1-[1,2]-1-5/gpononu gem 505 gtp 1 downlink vlan 505 tagged
To Abort the operation enter Ctrl-C
Adding bridge on 1-1-1-5/gpononu
Created bridge-interface-record 1-1-1-505-gponport-505/bridge
Adding bridge on 1-2-1-5/gpononu
Created bridge-interface-record 1-2-1-505-gponport-505/bridge

```

- OLT port IDs in a range

This example specifies the slot ID to 1, OLT port ID to 1,2,3,4, ONU ID to 4, GEM port ID to 504.

```

zSH> bridge add 1-1-[1-4]-4/gpononu gem 504 gtp 1 downlink vlan 504 tagged
To Abort the operation enter Ctrl-C
Adding bridge on 1-1-1-4/gpononu
Created bridge-interface-record 1-1-1-504-gponport-504/bridge
Adding bridge on 1-1-2-4/gpononu
Created bridge-interface-record 1-1-2-504-gponport-504/bridge
Adding bridge on 1-1-3-4/gpononu
Created bridge-interface-record 1-1-3-504-gponport-504/bridge
Adding bridge on 1-1-4-4/gpononu
Created bridge-interface-record 1-1-4-504-gponport-504/bridge

```

- GEM port IDs in a range

Note that to specify GEM port ID and ONU ID in a range, the **bridge add** command with *shelfID-slotID-OLTportID-GEMportID/gponport* format must be used.

This example specifies the slot ID to 1, OLT port ID to 2, GEM port IDs to 902 and 921, ONU ID is implied to 2 and 21, and Ethernet UNI port to 1 and 2.

```
zSH> bridge add 1-1-1-[902,921]/gponport gtp 1 downlink vlan 3811 tagged eth
[1-2] rg-brouted
```

To Abort the operation enter Ctrl-C

Adding bridge on 1-1-1-902/gponport

Created bridge-interface-record 1-1-1-902-gponport-3811/bridge

CPE Connection 1-1-1-902/gponport/1/1/0/0 has been created

CPE Connection 1-1-1-902/gponport/1/2/0/0 has been created

Adding bridge on 1-1-1-921/gponport

Created bridge-interface-record 1-1-1-921-gponport-3811/bridge

CPE Connection 1-1-1-921/gponport/1/1/0/0 has been created

CPE Connection 1-1-1-921/gponport/1/2/0/0 has been created

- Ethernet UNI ports in a range

This example specifies the slot ID to 1, OLT port ID to 1, ONU ID to 63, GEM port to 306, and Ethernet UNI port to 1 and 2.

```
zSH> bridge add 1-1-1-63/gpononu gem 306 gtp 1 downlink vlan 3811 tagged eth
[1,2]
```

Adding bridge on 1-1-1-63/gpononu

Created bridge-interface-record 1-1-1-306-gponport-3811/bridge

CPE Connection 1-1-1-306/gponport/1/1/0/0 has been created

CPE Connection 1-1-1-306/gponport/1/2/0/0 has been created

- All Ethernet UNI ports and WLAN UNI ports

This example specifies the slot ID to 1, OLT port ID to 1, GEM port ID to 363, ONU ID to 63, WLAN UNI port to all and Ethernet UNI port to all.

```
zSH> bridge add 1-1-1-63/gpononu gem 306 gtp 1 downlink vlan 3811 tagged eth
all wlan all rg-brouted
```

Adding bridge on 1-1-1-63/gpononu

Created bridge-interface-record 1-1-1-306-gponport-3811/bridge

CPE Connection 1-1-1-306/gponport/1/1/0/0 has been created

CPE Connection 1-1-1-306/gponport/1/2/0/0 has been created

CPE Connection 1-1-1-306/gponport/1/3/0/0 has been created

CPE Connection 1-1-1-306/gponport/1/4/0/0 has been created

CPE Connection 1-1-1-306/gponport/18/1/0/0 has been created

CPE Connection 1-1-1-306/gponport/18/2/0/0 has been created

CPE Connection 1-1-1-306/gponport/18/3/0/0 has been created

CPE Connection 1-1-1-306/gponport/18/4/0/0 has been created

Adding bridges with multiple interface ranges

- 1 Add bridges in RG mode with multiple interface ranges including UNI range.


```
zSH> bridge add 1-1-[1-3]-[301-304]/gponport gtp 1 downlink vlan 100 tagged eth
[1-2] rg-brouted
```

To Abort the operation enter Ctrl-C

Adding bridge on 1-1-1-301/gponport

Created bridge-interface-record 1-1-1-301-gponport-100/bridge

CPE Connection 1-1-1-301/gponport/1/1/0/0 has been created

CPE Connection 1-1-1-301/gponport/1/2/0/0 has been created

Adding bridge on 1-1-1-302/gponport

Created bridge-interface-record 1-1-1-302-gponport-100/bridge

CPE Connection 1-1-1-302/gponport/1/1/0/0 has been created

CPE Connection 1-1-1-302/gponport/1/2/0/0 has been created

Adding bridge on 1-1-1-303/gponport

Created bridge-interface-record 1-1-1-303-gponport-100/bridge

CPE Connection 1-1-1-303/gponport/1/1/0/0 has been created

CPE Connection 1-1-1-303/gponport/1/2/0/0 has been created

Adding bridge on 1-1-1-304/gponport

Created bridge-interface-record 1-1-1-304-gponport-100/bridge

CPE Connection 1-1-1-304/gponport/1/1/0/0 has been created

CPE Connection 1-1-1-304/gponport/1/2/0/0 has been created

Adding bridge on 1-1-2-301/gponport

Created bridge-interface-record 1-1-2-301-gponport-100/bridge

CPE Connection 1-1-2-301/gponport/1/1/0/0 has been created

CPE Connection 1-1-2-301/gponport/1/2/0/0 has been created

Adding bridge on 1-1-2-302/gponport

Created bridge-interface-record 1-1-2-302-gponport-100/bridge

CPE Connection 1-1-2-302/gponport/1/1/0/0 has been created

CPE Connection 1-1-2-302/gponport/1/2/0/0 has been created

Adding bridge on 1-1-2-303/gponport

Created bridge-interface-record 1-1-2-303-gponport-100/bridge

CPE Connection 1-1-2-303/gponport/1/1/0/0 has been created

CPE Connection 1-1-2-303/gponport/1/2/0/0 has been created

Adding bridge on 1-1-2-304/gponport

Created bridge-interface-record 1-1-2-304-gponport-100/bridge

CPE Connection 1-1-2-304/gponport/1/1/0/0 has been created

CPE Connection 1-1-2-304/gponport/1/2/0/0 has been created

Adding bridge on 1-1-3-301/gponport

Created bridge-interface-record 1-1-3-301-gponport-100/bridge

CPE Connection 1-1-3-301/gponport/1/1/0/0 has been created

CPE Connection 1-1-3-301/gponport/1/2/0/0 has been created

Adding bridge on 1-1-3-302/gponport

Created bridge-interface-record 1-1-3-302-gponport-100/bridge

CPE Connection 1-1-3-302/gponport/1/1/0/0 has been created

CPE Connection 1-1-3-302/gponport/1/2/0/0 has been created

Adding bridge on 1-1-3-303/gponport

Created bridge-interface-record 1-1-3-303-gponport-100/bridge

CPE Connection 1-1-3-303/gponport/1/1/0/0 has been created

CPE Connection 1-1-3-303/gponport/1/2/0/0 has been created

Adding bridge on 1-1-3-304/gponport

Created bridge-interface-record 1-1-3-304-gponport-100/bridge

CPE Connection 1-1-3-304/gponport/1/1/0/0 has been created

CPE Connection 1-1-3-304/gponport/1/2/0/0 has been created

2 View the created bridges.

zSH> **bridge show vlan 100**

Type	Orig	VLAN/SLAN	VLAN/SLAN	Physical	Bridge
St	Table	Data			

dwn		Tagged	100	1/1/1/1/gpononu	1-1-1-301-gponport-100/
bridge	UP				
dwn		Tagged	100	1/1/1/2/gpononu	1-1-1-302-gponport-100/
bridge	UP				
dwn		Tagged	100	1/1/1/3/gpononu	1-1-1-303-gponport-100/
bridge	DWN				
dwn		Tagged	100	1/1/1/4/gpononu	1-1-1-304-gponport-100/
bridge	UP				
dwn		Tagged	100	1/1/2/1/gpononu	1-1-2-301-gponport-100/
bridge	DWN				
dwn		Tagged	100	1/1/2/2/gpononu	1-1-2-302-gponport-100/
bridge	DWN				
dwn		Tagged	100	1/1/2/3/gpononu	1-1-2-303-gponport-100/
bridge	DWN				
dwn		Tagged	100	1/1/2/4/gpononu	1-1-2-304-gponport-100/
bridge	DWN				
dwn		Tagged	100	1/1/3/1/gpononu	1-1-3-301-gponport-100/
bridge	DWN				
dwn		Tagged	100	1/1/3/2/gpononu	1-1-3-302-gponport-100/
bridge	DWN				
dwn		Tagged	100	1/1/3/3/gpononu	1-1-3-303-gponport-100/
bridge	DWN				
dwn		Tagged	100	1/1/3/4/gpononu	1-1-3-304-gponport-100/
bridge	DWN				
12 Bridge Interfaces displayed					

3 View the created CPE connections.

zSH> **bridge show onu vlan 100**

ONU	GEM	ONU	DSCP	ONU UNI	OLT	OLT			
OLT Bridge	Port	UNI	to COS	VLAN/SLAN	VLAN/SLAN	G-VLAN	MVR	Service	Rg-Mode
ST									

1/3/1	301	eth 1	-----	----/----	Tagged	100	----	----	data
1-1-3-301-gponport-100/bridge					DWN				B-Routed
1/3/1	301	eth 2	-----	----/----	Tagged	100	----	----	data
1-1-3-301-gponport-100/bridge					DWN				B-Routed
1/2/3	303	eth 1	-----	----/----	Tagged	100	----	----	data
1-1-2-303-gponport-100/bridge					DWN				B-Routed
1/2/3	303	eth 2	-----	----/----	Tagged	100	----	----	data
1-1-2-303-gponport-100/bridge					DWN				B-Routed
1/3/4	304	eth 1	-----	----/----	Tagged	100	----	----	data
1-1-3-304-gponport-100/bridge					DWN				B-Routed
1/3/4	304	eth 2	-----	----/----	Tagged	100	----	----	data
1-1-3-304-gponport-100/bridge					DWN				B-Routed

```

1/1/2      302  eth 1  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-1-302-gponport-100/bridge          DWN
1/1/2      302  eth 2  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-1-302-gponport-100/bridge          DWN
1/2/2      302  eth 1  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-2-302-gponport-100/bridge          DWN
1/2/2      302  eth 2  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-2-302-gponport-100/bridge          DWN
1/3/2      302  eth 1  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-3-302-gponport-100/bridge          DWN
1/3/2      302  eth 2  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-3-302-gponport-100/bridge          DWN
1/3/3      303  eth 1  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-3-303-gponport-100/bridge          DWN
1/3/3      303  eth 2  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-3-303-gponport-100/bridge          DWN
1/1/4      304  eth 1  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-1-304-gponport-100/bridge          UP
1/1/4      304  eth 2  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-1-304-gponport-100/bridge          UP
1/2/1      301  eth 1  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-2-301-gponport-100/bridge          DWN
1/2/1      301  eth 2  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-2-301-gponport-100/bridge          DWN
1/2/4      304  eth 1  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-2-304-gponport-100/bridge          DWN
1/2/4      304  eth 2  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-2-304-gponport-100/bridge          DWN
1/1/3      303  eth 1  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-1-303-gponport-100/bridge          DWN
1/1/3      303  eth 2  ----- /---- Tagged 100  ----  ---- data  B-Routed
1-1-1-303-gponport-100/bridge          DWN
12 Bridge Interfaces displayed
24 GPON ONU Connections displayed

```

4 View CPE.

```
zSH> cpe show 1/1/1
```

```
CPE 1/1/1
```

```
Service: DATA
```

GEM	UNI	UNI-VLAN/SLAN	VLAN/SLAN (COS,VID)	G-VLAN	Admin	Oper	Rg-Mode
301	eth 1		0,100/----	0	up		B-Routed
301	eth 2		0,100/----	0	up		B-Routed

Deleting bridges with multiple interface ranges

Delete bridges with multiple interface ranges including UNI range.

Note that if users have more than one CPE connection associated with those bridges users want to delete, users can either delete the CPE connection profiles separately or use the “all” command line argument.

- 1 Delete all CPE connections associated with Ethernet 1:

```

zSH> bridge delete 1-1-[1-3]-[301-304]/gponport eth 1
To Abort the operation enter Ctrl-C
CPE Connection 1-1-1-301/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-1-302/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-1-303/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-1-304/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-2-301/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-2-302/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-2-303/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-2-304/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-3-301/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-3-302/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-3-303/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-3-304/gponport/1/1/0/0 has been deleted
0 bridge interfaces deleted out of 12 found

```

2 Delete bridge interfaces and all associated CPE connections:

```

zSH> bridge delete 1-1-[1-3]-[301-304]/gponport all
To Abort the operation enter Ctrl-C
CPE Connection 1-1-1-301/gponport/1/2/0/0 has been deleted
1-1-1-301/gponport delete complete
CPE Connection 1-1-1-302/gponport/1/2/0/0 has been deleted
1-1-1-302/gponport delete complete
CPE Connection 1-1-1-303/gponport/1/2/0/0 has been deleted
1-1-1-303/gponport delete complete
CPE Connection 1-1-1-304/gponport/1/2/0/0 has been deleted
1-1-1-304/gponport delete complete
CPE Connection 1-1-2-301/gponport/1/2/0/0 has been deleted
1-1-2-301/gponport delete complete
CPE Connection 1-1-2-302/gponport/1/2/0/0 has been deleted
1-1-2-302/gponport delete complete
CPE Connection 1-1-2-303/gponport/1/2/0/0 has been deleted
1-1-2-303/gponport delete complete
CPE Connection 1-1-2-304/gponport/1/2/0/0 has been deleted
1-1-2-304/gponport delete complete
CPE Connection 1-1-3-301/gponport/1/2/0/0 has been deleted
1-1-3-301/gponport delete complete
CPE Connection 1-1-3-302/gponport/1/2/0/0 has been deleted
1-1-3-302/gponport delete complete
CPE Connection 1-1-3-303/gponport/1/2/0/0 has been deleted
1-1-3-303/gponport delete complete
CPE Connection 1-1-3-304/gponport/1/2/0/0 has been deleted
1-1-3-304/gponport delete complete
12 bridge interfaces deleted out of 12 found

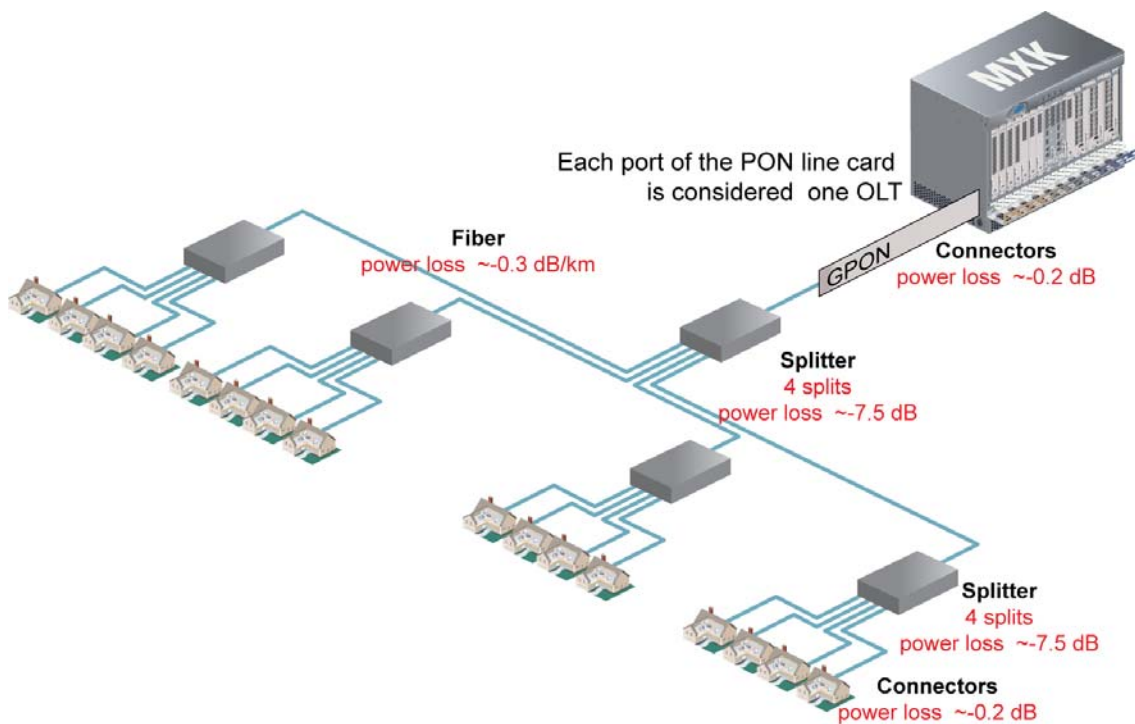
```

Planning GPON networks

When deploying GPON networks, users have to think in optical terms, rather than electrical or copper based terms. With copper based solutions users think of distance and transport technology (“Will ADSL or VDSL reach from the CO to the subscribers?” is a significant network design question); with fiber based networks, and GPON in particular, users have to think in terms of optical link power loss budgets.

Link loss is the amount of signal attenuation as users proceed farther away from the OLT toward the subscribers’ ONTs. Each component, including the fiber cable itself, degrades the signal. Attenuation is the term used for describing the amount of signal degradation.

Figure 106: Link loss in an GPON Optical Deployment Network



The plan for both a GPON network and Active Ethernet network should include a link loss budget map that shows how each component, even the distance of each length of fiber, should affect signal attenuation. Because GPON lines are split into multiple lines which have a significant power loss, the link loss budget map is a more important requirement for GPON.



Note: The power loss may vary by manufacturer, refer to equipment vendor for the detail.

Component	Loss
Optical fiber	-0.3 dB per kilometer

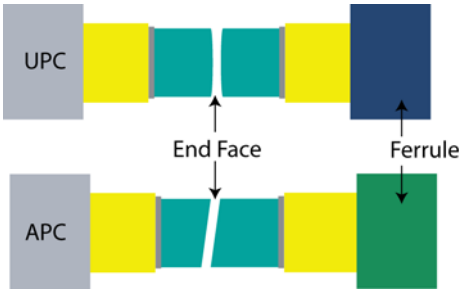
Component	Loss
Splitters	The link loss for splitters depends on the number of splits • 2 splits, -4 dB • 4 splits, -7.5 dB • 8 splits, -11 dB • 16 splits, -14 dB • 32 splits, -18 dB • 64 splits, -21.5 dB
Splices	-0.1 dB
Connectors	-0.2 dB
Couplers	Couplers are connectorized means for splicing cable. -0.4 dB

Installation testing

The theoretical link loss budget map is very important when installing fiber. Testing should be done before and after each component is added. Matching the actual signal attenuation with the theoretical link loss budget map helps identify problems such as

- macro bends in cables (too small a bend radius)
- connector loss from back reflection (the contact between the face ends of fiber in a connector, or a splice)
- incorrectly matching UPC and APC connectors may also create back reflections. UPC connectors (Ultra Physical Contact) have a slightly spherical end face. APC connectors (Angled Physical Contact) use an industry standard angle on the end face of the fiber. (Though users should be aware of older, non standard APC connectors which use a different angle.)

Figure 107: End face of UPC and APC connectors



There are testing tools on the market which can be used to test the components as added.

The actual figures that are discovered during installation testing should also be noted and filed as they may also be helpful when troubleshooting problems which may arise in the ODN in the future.

Handling fiber

Handling of fiber requires special precautions for those familiar with copper wiring.



WARNING!

Never look into an active optical fiber. Exposure to invisible LASER radiation may cause serious retinal damage or even blindness.



WARNING! Clean hands after handling optical fibers. Small pieces of glass are not always visible and can cause eye damage.

Get medical assistance immediately for any glass that comes into contact with your eye.

Fiber needs to be kept clean. Contaminants may obstruct the passing of light. Notable contaminants include

- oil from hands
- dust particles
- lint
- the residue which may be left when using wet cleaning methods
- scratches which may be from dry cleaning methods or mishandling fiber.

Fiber requires a handling discipline which includes

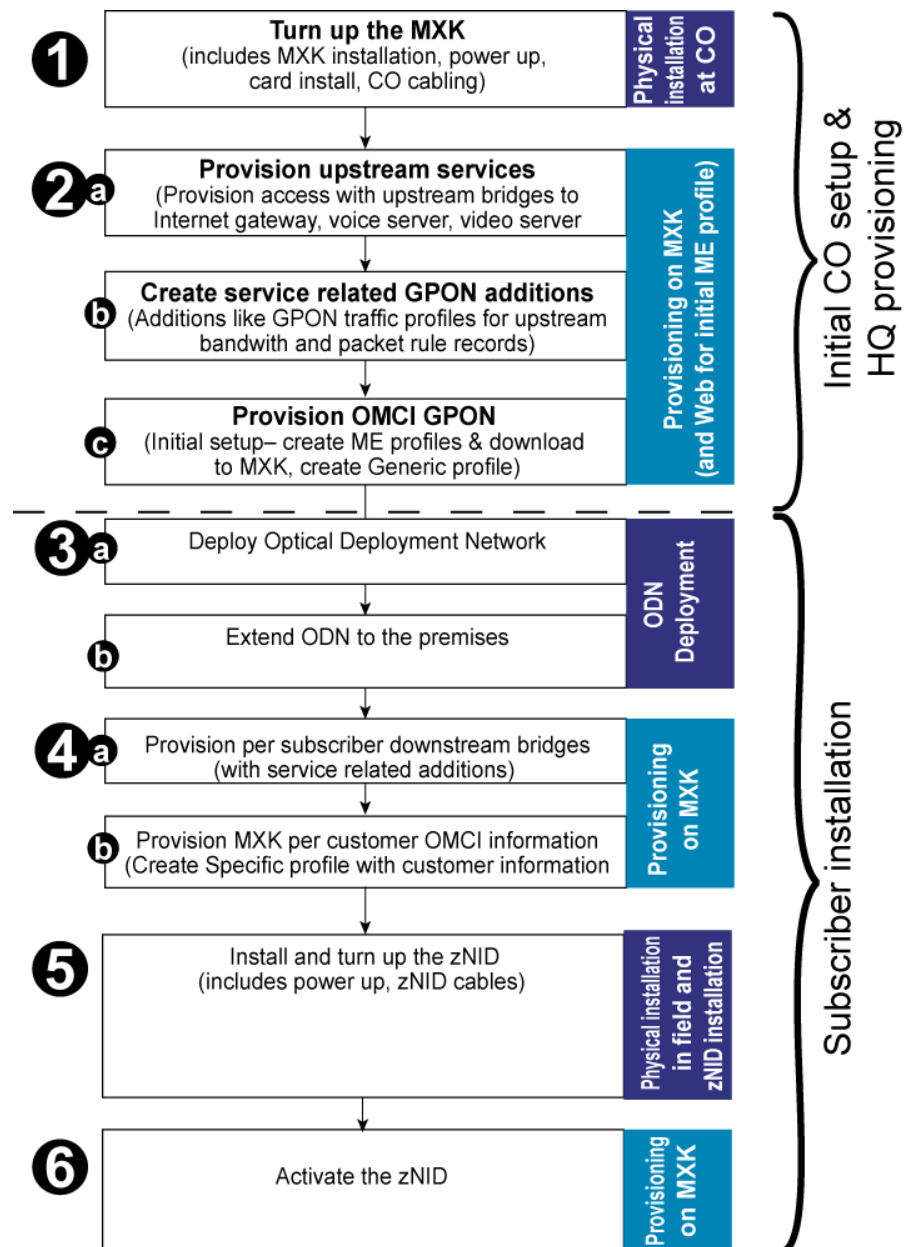
- inspecting fiber ends (with a fiber inspection probe)
- cleaning fiber, with either a wet cleaning method, dry cleaning method or both.
- fiber cannot be bent too far. Bending fiber too far will keep the optical signal from bending. Users may see the light through the sheathing of the cable. These microbends may also create microfractures in the glass of the fiber resulting in signal loss.

Smart OMCI GPON zNID installation

This section includes the following topics:

- [OMCI overview, page 693](#)
- [Smart OMCI overview, page 693](#)
- [OMCI GPON zNID installation with Smart OMCI, page 696](#)
- [Delete the OMCI profile, page 717](#)
- [Import and export the OMCI profile, page 720](#)

Figure 108: Installation procedure for OMCI GPON zNID with Smart OMCI



OMCI overview

The ONT Management and Control Interface (OMCI) is a protocol defined by ITU-T G.988 that enables the Optical Line Terminal (OLT) to control an Optical Network Unit (ONU). This protocol allows the OLT to:

- Establish and release connections across the ONU.
- Manage the User Network Interfaces (UNIs) at the ONU.
- Request configuration information and performance statistics.
- Autonomously inform the system operator of events such as link failures.

The OMCI protocol runs across the GEM connection between the OLT controller and the ONU controller that is established at ONU initialization.

The ONU management and control interface requirements given in the ITU-T G.988. Recommendation are needed to manage ONU in the following areas:

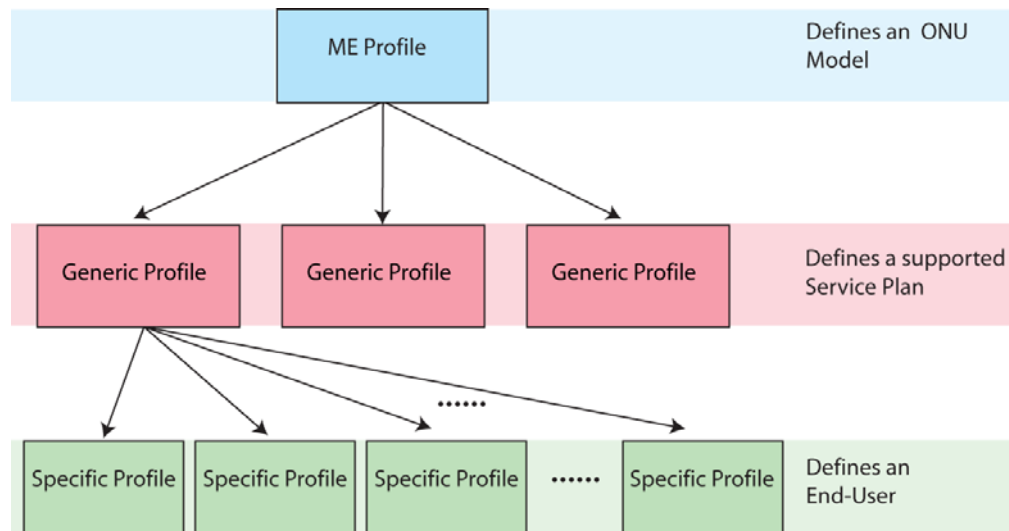
- Configuration management
- Fault management
- Performance management
- Security management

Smart OMCI overview

OMCI Profiles

Smart OMCI functionality is implemented on the MXK by using OMCI profiles.

The three types of OMCI profiles defined in the system are ME, Generic, and Specific. Each profile type is synonymous to a task performed in the network deployment phase. As shown in the [Figure 109](#), these three profile types have a hierarchical relationship.

Figure 109: Smart OMCI Architecture

- ME profile

The ME profile defines an ONU model and a service profile.

The ME profile contains all the information required to support an ONU and defines the OMCI commands that OLT uses to configure an ONU. If a service provider supports 3 different ONUs in their network, there will be 3 ME profiles in the MXK. The ME profile is created on the MXK by an ME profile file that is downloaded from Zhone's website.

- Generic profile

The Generic profile defines the common default parameters for service plan supported by the service provider for a given ONU model.

A Generic profile is always associated with only one ME profile and contains the values for network parameters that define a service plan and the value for infrastructure network elements such as the softswitch IP address. If the service provider supports 5 different service plans on each of the 3 supported ONU models, there will be a total of 15 Generic Profiles in the MXK (5 Generic profiles for each of the ME profile). The Generic Profile can be created using the CLI, ZMS or WebUI. The ME profile and Generic profile are created at the time of initial network deployment before activating the user.

- Specific profile

The Specific profile give values to parameters per user based before activating the end-user. The Specific profile is always associated with only one Generic profile. The Specific profile contains value for specific users, and the variable list in the Specific profile is same as in the Generic profile. At creation, the Specific profile automatically inherits all the values of the parent Generic profile and does not require modification when the same values are used. When there is user specific information,

like a telephone number, the values can be overridden by modifying those variables in the Specific profile. The variables defined in the Generic and Specific profiles are values used by the OMCI commands in the ME profile.

When activating an end-user, based on the service plan and ONU model being used by the end-user, choose the appropriate ME profile and Generic profile to associate with the Specific profile.

The order of precedence for implementing a value in the information field that is sent to the ONU is first the Specific profile, then the Generic profile, and finally the ME profile.

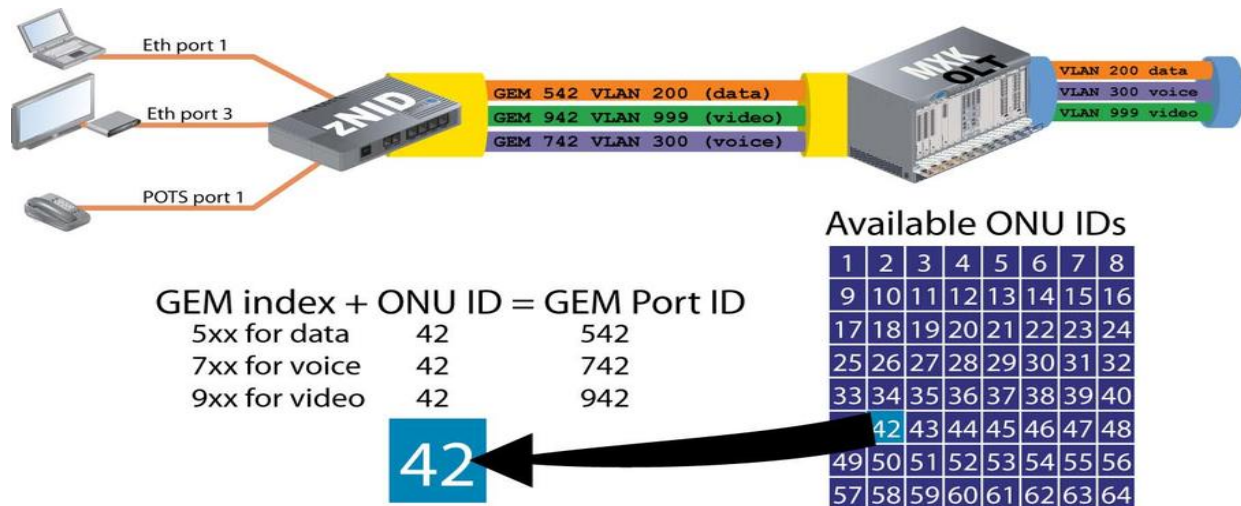
ME profiles and Generic profiles are normally created by a network analyst or network architect. The ME profile is the profile of the capabilities of the ONU model. Multiple MEs may be used for a single model. The more common strategy is to have all attributes for the ONU model configured in the ME profile. The Generic profile is intended to define ISP user bundles. If the ME profile has all ports configured, the Generic profile may define which are active for the end user. The specific profile is the end user profile and contains end user specific information, such as the phone number.

Dynamic GEM ports

If an ONU is planned to be managed with Smart OMCI, when users create a GEM port, make sure the GEM port ID = GEM index + ONU ID, where GEM index is the port offset selected in the Smart OMCI web interface, from 5xx to 35xx.

Each ONU supports up to 16 GPON GEM ports.

Figure 110: Dynamic GEM port ID are created from the GEM index and the ONU ID



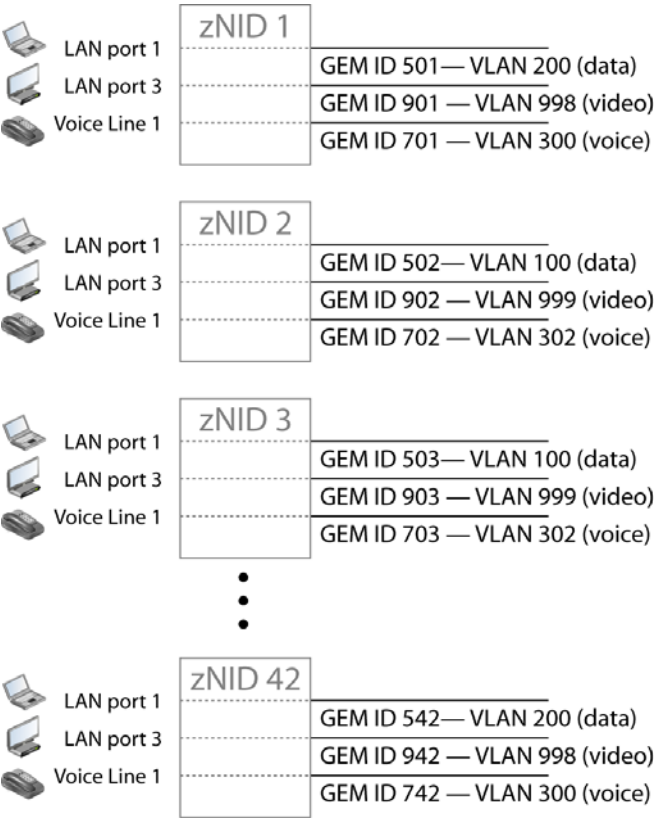
When creating downstream services on the MXK, the subport information in the **bridge add** command would be the same as the GEM port ID.

```
zSH> bridge add 1-4-4-542/gponport gtp 1 downlink vlan 200 tagged
```

```
Adding bridge on 1-4-4-542/gponport
Created bridge-interface-record 1-4-4-542-gponport-200/bridge
```

In the above example, GEM port 1-4-4-542 has been created on ONU 1-4-4-42/gpononu. The GEM port ID, 542, is the sub-port for the **bridge add** command, and it is in the **bridge add** command which defines which VLAN is matched to the GEM port.

Figure 111: zNID 1 and 42 are from the same company. zNID 2 and 3 are from separate residences



OMCI GPON zNID installation with Smart OMCI

Generally these are the steps to follow to configure the MXK to be able to manage OMCI GPON zNID with Smart OMCI:

- [Create a ME profile through SMART OMCI web-interface, page 697](#)
- [Download a ME profile file to the MXK, page 701](#)
- [Create a ME profile for the selected ONT model, page 702](#)
- [Create Generic profiles for service plan, page 702](#)
- [Create high speed Internet on GPON OMCI on uplink and downlink bridges, page 706](#)

- [Create uplink and downlink bridges on GPON OMCI for video, page 710](#)
- [Create uplink and downlink bridge on GPON OMCI for VoIP, page 713](#)

Create a ME profile through SMART OMCI web-interface

Zhone Technologies provides the service provider a Smart OMCI web-interface to select desired ONU model and services.

Creating ME profile file through Smart OMCI web-interface

Using the Smart OMCI web-interface the service provider creates the ME profile file that containing the ME structure information which is unique to the ONU hardware model.

Access to the Smart OMCI web-interface can be through Zhone's website.

To create an ME profile file:

- 1 Navigate to the Zhone website at "http://www.zhone.com/support/tools/omci/".
- 2 Access the website by entering the email address and the password selected at registration.

Note: skip this step if users are already signed in.



- 3 Select desired ONU model, then click **Continue**.

This example selects ONU model ZNID-GPON-2510.



After selecting the ONU model, the Smart OMCI web-interface updates to display the list of services that are supported on this ONU hardware model.

- 4 Select the desired services. For each service, users can select the supported physical interfaces, GEM Index, and VLAN filtering.

GEM index is in the range of 5xx to 35xx.

This example selects GEM index 5xx for data service on port *eth1* and *eth2*, GEM index 7xx for voice service on port *POTS1* and *POTS2*, GEM index 9xx for video service on port *eth3* and *eth4*.



Note: Take a note of the GEM index users selected for different services. It could be used to calculate the GEM port ID with the following formula:

GEM port ID = GEM index + ONU ID

The GEM port ID is used when users provisioning services on bridges or routers by using the **bridge add** commands.

Refer to [Create a GEM port on page 938](#) for configuration information.

Zone | Support | Smart OMCI: Data Config - Microsoft Internet Explorer

Address: <http://www.zhone.com/support/tools/omci/?pn=ZNID-GPON-2510>

ZNID-GPON-2510: Indoor GPON ONT, 4xFE, 2xPOTS

Zhone's ZNID-GPON-2510 is an indoor ONT designed for high performance applications. It features 4 10/100Mbps LAN Ports and 2 POTS Voice FXS ports in a compact enclosure for indoor use. 2510 may be installed as a desktop or, wall mounted and is powered via an AC Adaptor. 2510 ONT is ideal for triple-play service deployments in Fiber-to-the-Home/Premise application. Industry standard SIP

■ [ZNID-GPON-2510 Datasheet](#)

Data & Video Configuration

☒ **Port ETH 1** GEM: 5xx Data Add single tag to untagged frame
VLAN ID: 100 COS: 0 (802.1p)

☒ **Port ETH 2** GEM: 5xx Data Add single tag to untagged frame
VLAN ID: 100 COS: 0 (802.1p)

☒ **Port ETH 3** GEM: 9xx Video Add single tag to untagged frame
VLAN ID: 999 COS: 4 (802.1p)

☒ **Port ETH 4** GEM: 9xx Video -choose data service-

Voice Configuration

☒ **Port POTS 1**

☒ **Port POTS 2**

GEM 7xx VLAN ID 200 COS 7 (802.1p)

Country Code US

Host IP Option Static

Netmask 0.0.0.0

Gateway 0.0.0.0

Protocol SIP

VOIP Server 0.0.0.0

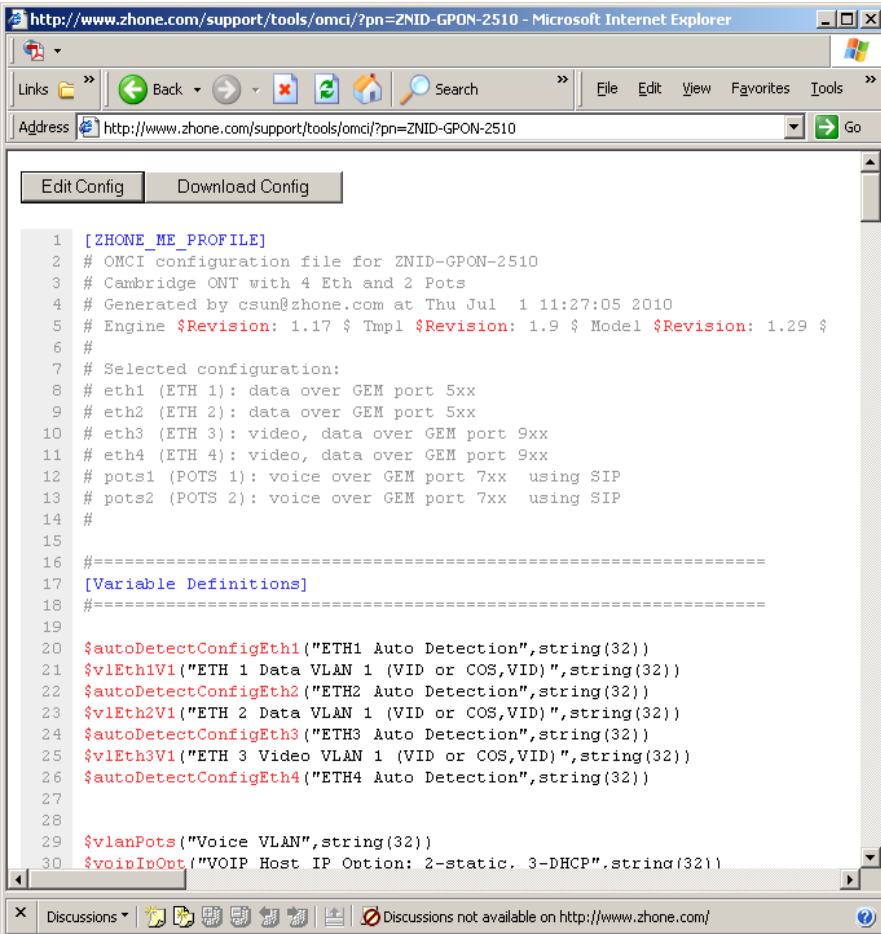
Primary DNS 0.0.0.0

Secondary DNS 0.0.0.0

SIP Domain

Discussions Discussions not available on <http://www.zhone.com/>

- 5 Click the **Create Configuration File** button. An ME profile file is created and displayed in the ME profile file page.



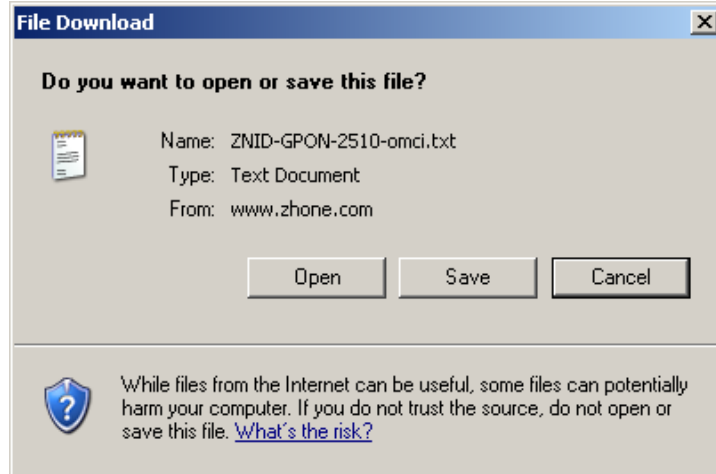
```

1  [ZONE_ME_PROFILE]
2  # OMCI configuration file for ZNID-GPON-2510
3  # Cambridge ONT with 4 Eth and 2 Pots
4  # Generated by csun@zhone.com at Thu Jul  1 11:27:05 2010
5  # Engine $Revision: 1.17 $ Tmpl $Revision: 1.9 $ Model $Revision: 1.29 $
6  #
7  # Selected configuration:
8  # eth1 (ETH 1): data over GEM port 5xx
9  # eth2 (ETH 2): data over GEM port 5xx
10 # eth3 (ETH 3): video, data over GEM port 9xx
11 # eth4 (ETH 4): video, data over GEM port 9xx
12 # pots1 (POTS 1): voice over GEM port 7xx using SIP
13 # pots2 (POTS 2): voice over GEM port 7xx using SIP
14 #
15
16 #=====
17 [Variable Definitions]
18 #=====
19
20 $autoDetectConfigEth1("ETH1 Auto Detection",string(32))
21 $v1Eth1V1("ETH 1 Data VLAN 1 (VID or COS,VID)",string(32))
22 $autoDetectConfigEth2("ETH2 Auto Detection",string(32))
23 $v1Eth2V1("ETH 2 Data VLAN 1 (VID or COS,VID)",string(32))
24 $autoDetectConfigEth3("ETH3 Auto Detection",string(32))
25 $v1Eth3V1("ETH 3 Video VLAN 1 (VID or COS,VID)",string(32))
26 $autoDetectConfigEth4("ETH4 Auto Detection",string(32))
27
28
29 $vlanPots("Voice VLAN",string(32))
30 $voipIpOut("VOIP Host IP Option: 2-static, 3-DHCP",string(32))

```

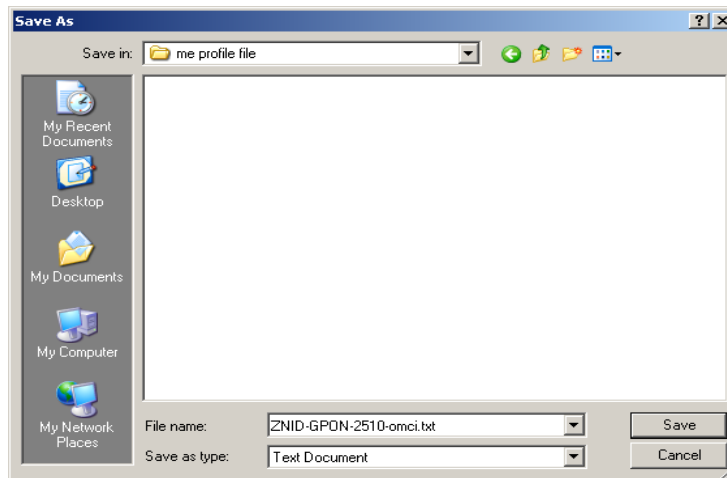
- 6 Two options are displayed on the top of the ME profile file page, **Edit Config** and **Download Config**.
- Clicking on the **Edit Config** button causes the web-interface to return to the service page. This page lists the current selection. Users can change the configuration, and create a new ME profile file.

- Clicking on the **Download Config** button causes the web-interface to display a File Download window.



Click **Save** to open the Save As window.

In the Save As window, changes the path and chooses the appropriate file name to save the newly created ME profile file. The file type must be text (.txt).



Download a ME profile file to the MXK

Downloading ME profile file to MXK

The ME profile file must be downloaded from a TFTP/ SFTP server to MXK.

- 1 Verify that the current directory is the root (i.e. /card1) with the **pwd** command. If not in this directory use the **cd** (change directory) command to move to it.

```
zSH> pwd
/card1
```

- 2 Create a directory at the root level (i.e. /card1), then download the ME profile file.

In this example the directory is named as *me*.

There are no restrictions on the directory name.

```
zSH> mkdir me
```

- 3 Move to the newly created directory.

```
zSH> cd me
```

- 4 Download the ME profile file to the current directory in the MXK with the **file download** command. This example downloads the ME profile file *ZNID-GPON-2510-omci.txt* from a TFTP server *172.16.80.201* to the MXK */me* directory, and save the ME profile file with the same name.

```
zSH> file download 172.16.80.201 /
ZNID-GPON-2510-omci.txt ZNID-GPON-2510-omci.txt
Bytes copied:      18411
File download successful
```

Create a ME profile for the selected ONT model

The software supports a text import capability to read the ME profile file and learn the ME structure of the new ONU. The ME profile contains OMCI ME commands.

Creating ME profile for selected ONT model

Create an ME profile from the ME profile file. One ME profile is created for each ONU model.

- 1 Create an ME profile. This example creates a ME profile from the downloaded ME profile file *ZNID-GPON-2510-omci.txt*, and name it to *2510-tripleplay-me*.

```
zSH> gpononu profile create me 2510-tripleplay-me
ZNID-GPON-2510-omci.txt
Profile created
```

- 2 Verify the created ME profile name:

```
zSH> gpononu profile show me
2510-tripleplay-me
```

```
.....
```

Create Generic profiles for service plan

The Generic profile defines the values of variables that define service plans. It also contains values of system variables. The system values, service plan values are entered by the service provider as part of system commissioning.

If the service provider intend to offer 3 different service plans that are supported on 5 different ONU hardware models, service provider should create 5 ME profiles and 15 Generic profiles in the system.

Creating Generic profiles for service plan

To create a Generic profile:

1 Create a Generic profile:

```
zSH> gpononu profile create gen 2510-tripleplay-gen
2510-tripleplay-me
Profile created
```

2 Verify the created Generic profile name.

```
zSH> gpononu profile show gen
2510-tripleplay-gen
```

3 Update the Generic profile.

To assign or change a parameter, enter the line number, click Enter, then enter the value, at last enter s to save the profile.

```
zSH> gpononu profile update gen 2510-tripleplay-gen
Generic Profile: 2510-tripleplay-gen
1  "ETH1 Auto Detection [0]"
2  "ETH 1 Data VLAN 1 (VID or COS,VID) [0,100]"
3  "ETH 1 Data VLAN 2 (VID or COS,VID) [0,100]"
4  "ETH2 Auto Detection [0]"
5  "ETH 2 Data VLAN 1 (VID or COS,VID) [0,100]"
6  "ETH 2 Data VLAN 2 (VID or COS,VID) [0,100]"
7  "ETH3 Auto Detection [0]"
8  "ETH 3 Video VLAN 1 (VID or COS,VID) [4,999]"
9  "ETH 3 Video VLAN 2 (VID or COS,VID) [4,999]"
10 "ETH4 Auto Detection [0]"
11 "Voice VLAN [7,200]"
12 "VOIP Host IP Option: 2-static, 3-DHCP [2]"
13 "VOIP Host IP [0.0.0.0]"
14 "VOIP Netmask [0.0.0.0]"
15 "VOIP Gateway [0.0.0.0]"
16 "VOIP Server IP [0.0.0.0]"
17 "VOIP Primary DNS [0.0.0.0]"
18 "VOIP Secondary DNS [0.0.0.0]"
19 "Country Code [ 1]"
20 "Rx Gain [0]"
21 "Tx Gain [0]"
22 "Out-of-band DTMF [0]"
23 "Echo Cancel: 1-enable, 0-disable [1]"
24 "POTS1 Dial Number [1111]"
25 "POTS1 User Name [11111]"
26 "POTS1 Password [11111]"
27 "POTS2 Dial Number [2222]"
28 "POTS2 User Name [22222]"
29 "POTS2 Password [22222]"
```

```

30  "Fax Mode [0]"
31  "CID Features [63]"
32  "Call Waiting Features [3]"
33  "Call Progress or Transfer Features [255]"
34  "Call Present Features [15]"
35  "ETH 1 Admin Status: 0-Up, 1-Down [0]"
36  "ETH 2 Admin Status: 0-Up, 1-Down [0]"
37  "ETH 3 Admin Status: 0-Up, 1-Down [0]"
38  "ETH 4 Admin Status: 0-Up, 1-Down [0]"
39  "POTS 1 Admin Status: 0-Up, 1-Down [0]"
40  "POTS 2 Admin Status: 0-Up, 1-Down [0]"
Enter OMCI edit command or [s]ave, [q]uit, [h]elp:h

```

Available Commands:

```

E      - display edit data (short)
H      - display help
L      - display edit data (long)
Q      - quit without save
S      - save and exit
1..n   - edit variable #n

```

```

Enter OMCI edit command or [s]ave, [q]uit, [h]elp:2
"ETH 1 Data VLAN 1 (VID or COS,VID) [0,100]" : 100
Enter OMCI edit command or [s]ave, [q]uit, [h]elp:3
"ETH 1 Data VLAN 2 (VID or COS,VID) [0,100]" : 100
Enter OMCI edit command or [s]ave, [q]uit, [h]elp:20
"Rx Gain [0]" : 12
Enter OMCI edit command or [s]ave, [q]uit, [h]elp:21
"Tx Gain [0]" : 12
Enter OMCI edit command or [s]ave, [q]uit, [h]elp:30
"Fax Mode [0]" : 1
...
Enter OMCI edit command: s
GENERIC profile has been saved

```

- 4 View additional edit information for the variables in the Generic profile with the **gpononu profile update gen** command and enter OMCI edit command **L** (not case sensitive).

```

zSH> gpononu profile update gen 2510-tripleplay-gen
Generic Profile: 2510-tripleplay-gen
1  "ETH1 Auto Detection [0]"
2  "ETH 1 Data VLAN 1 (VID or COS,VID) [0,100]" 100
3  "ETH 1 Data VLAN 2 (VID or COS,VID) [0,100]" 100
4  "ETH2 Auto Detection [0]"
5  "ETH 2 Data VLAN 1 (VID or COS,VID) [0,100]"
6  "ETH 2 Data VLAN 2 (VID or COS,VID) [0,100]"
7  "ETH3 Auto Detection [0]"
8  "ETH 3 Video VLAN 1 (VID or COS,VID) [4,999]"
9  "ETH 3 Video VLAN 2 (VID or COS,VID) [4,999]"
10 "ETH4 Auto Detection [0]"
11 "Voice VLAN [7,200]"
12 "VOIP Host IP Option: 2-static, 3-DHCP [2]"
13 "VOIP Host IP [0.0.0.0]"

```

```

14 "VOIP Netmask [0.0.0.0]"
15 "VOIP Gateway [0.0.0.0]"
16 "VOIP Server IP [0.0.0.0]"
17 "VOIP Primary DNS [0.0.0.0]"
18 "VOIP Secondary DNS [0.0.0.0]"
19 "Country Code [ 1]"
20 "Rx Gain [12]"
21 "Tx Gain [12]"
22 "Out-of-band DTMF [0]"
23 "Echo Cancel: 1-enable, 0-disable [1]"
24 "POTS1 Dial Number [1111]"
25 "POTS1 User Name [1111]"
26 "POTS1 Password [1111]"
27 "POTS2 Dial Number [2222]"
28 "POTS2 User Name [2222]"
29 "POTS2 Password [2222]"
30 "Fax Mode [1]"
31 "CID Features [63]"
32 "Call Waiting Features [3]"
33 "Call Progress or Transfer Features [255]"
34 "Call Present Features [15]"
35 "ETH 1 Admin Status: 0-Up, 1-Down [0]"
36 "ETH 2 Admin Status: 0-Up, 1-Down [0]"
37 "ETH 3 Admin Status: 0-Up, 1-Down [0]"
38 "ETH 4 Admin Status: 0-Up, 1-Down [0]"
39 "POTS 1 Admin Status: 0-Up, 1-Down [0]"
40 "POTS 2 Admin Status: 0-Up, 1-Down [0]"
Enter OMCI edit command or [s]ave, [q]uit, [h]elp:1
ID    Generic Profile: 2510-tripleplay-gen
====
=====
=====
1  Name      : $autoDetectConfigEth1
   Comment   : ETH1 Auto Detection
   Type      : string(32)
   Gen Value :
   Default Value: 0
2  Name      : $vlEth1V1
   Comment   : ETH 1 Data VLAN 1 (VID or COS,VID)
   Type      : string(32)
   Gen Value : 100
   Default Value: 0,100
3  Name      : $autoDetectConfigEth2
   Comment   : ETH2 Auto Detection
   Type      : string(32)
   Gen Value : 100
   Default Value: 0,100
<SPACE> for next page, <CR> for next line, A for all, Q
to quitq

```

Create high speed Internet on GPON OMCI on uplink and downlink bridges

The high speed Internet application uses uplink and downlink bridges with a VLAN ID. Users should notice from the flowchart and procedures that provisioning video also uses uplink/downlink bridge configurations, just the GEM port setup (from the OMCI profile), GPON traffic profile and VLAN are different. For triple play services (As long as the OMCI profiles are configured properly) users can add the video bridge or VoIP bridge in the same process. For ease of discussion each of the applications is described separately in this chapter.

For data service users will create uplink/downlink bridges with VLAN 100.

Creating the GPON traffic profile

GPON traffic profiles are a template for defining how traffic will be handled on the bridge with which the GTP is associated. One GTP may be associated with many different bridges. The GTP in this procedure will create a high bandwidth configuration.

Refer to [Create a GEM port on page 938](#) and [Configure GPON traffic profile on page 925](#) to get detail configuration and parameter description.

The following is recommended for high speed data configurations.

```
zSH> new gpon-traffic-profile 1
gpon-traffic-profile 1
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}: 1024 in Kbps
traffic-class: -----> {ubr}: ubr is the default value
compensated: -----> {false}:
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: ----> {0}:
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Creating uplink and downlink bridges

Create an uplink and downlink bridge for VLAN 100:

1 Create the uplink bridge interface:

Add the bridge interface for the uplink.

Make sure VLAN ID matches the VLAN ID users assigned for data service in the Generic Profile. This example, data services uses VLAN 100.

```

zSH> bridge add 1-a-5-0/eth uplink vlan 100
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5/bridge
Bridge-path added successfully

```

2 Create downlink bridge interface:

Uses the GEM index assigned in the Smart OMCI web tool to calculate the GEM port ID with the following formula:

GEM port ID = GEM index + ONU ID

This example uses GEM index 5xx for data service, and ONT ID is 4/4/1, so the GEM port ID is 501.

```

zSH> bridge add 1-4-4-501/gponport gtp 1 downlink vlan 100 tagged
Adding bridge on 1-4-4-501/gponport
Created bridge-interface-record 1-4-4-501-gponport-100/bridge

```

Creating the Specific profile for a new user

On MXK create and modify Specific profile for each user; in the case of specific profiles, the OMCI supports are associated with the ONT.

Only one Specific profile can be added on an ONT.

To add a new user:

1 Create and modify the Specific profile.

- a** Create the Specific profile, selecting the ME profile and Generic profile to associate with the Specific (user) profile.

```

zSH> gpononu profile create spec 4/4/1
2510-tripleplay-me 2510-tripleplay-gen
Profile created

```

- b** Update the Specific profile.

```

zSH> gpononu profile update spec 4/4/1
Specific Profile: 4/4/1
1 "ETH1 Auto Detection [0]"
2 "ETH 1 Data VLAN 1 (VID or COS,VID) [0,100]" 100
3 "ETH 1 Data VLAN 2 (VID or COS,VID) [0,100]" 100
4 "ETH2 Auto Detection [0]"
5 "ETH 2 Data VLAN 1 (VID or COS,VID) [0,100]"
6 "ETH 2 Data VLAN 2 (VID or COS,VID) [0,100]"
7 "ETH3 Auto Detection [0]"
8 "ETH 3 Video VLAN 1 (VID or COS,VID) [4,999]"
9 "ETH 3 Video VLAN 2 (VID or COS,VID) [4,999]"
10 "ETH4 Auto Detection [0]"
11 "Voice VLAN [7, 200]"
12 "VOIP Host IP Option: 2-static, 3-DHCP [2]"
13 "VOIP Host IP [0.0.0.0]"
14 "VOIP Netmask [0.0.0.0]"
15 "VOIP Gateway [0.0.0.0]"

```

```

16 "VOIP Server IP [0.0.0.0]"
17 "VOIP Primary DNS [0.0.0.0]"
18 "VOIP Secondary DNS [0.0.0.0]"
19 "Country Code [ 1]"
20 "Rx Gain [12]"
21 "Tx Gain [12]"
22 "Out-of-band DTMF [0]"
23 "Echo Cancel: 1-enable, 0-disable [1]"
24 "POTS1 Dial Number [1111]"
25 "POTS1 User Name [1111]"
26 "POTS1 Password [1111]"
27 "POTS2 Dial Number [2222]"
28 "POTS2 User Name [2222]"
29 "POTS2 Password [2222]"
30 "Fax Mode [1]"
31 "CID Features [63]"
32 "Call Waiting Features [3]"
33 "Call Progress or Transfer Features [255]"
34 "Call Present Features [15]"
35 "ETH 1 Admin Status: 0-Up, 1-Down [0]"
36 "ETH 2 Admin Status: 0-Up, 1-Down [0]"
37 "ETH 3 Admin Status: 0-Up, 1-Down [0]"
38 "ETH 4 Admin Status: 0-Up, 1-Down [0]"
39 "POTS 1 Admin Status: 0-Up, 1-Down [0]"
40 "POTS 2 Admin Status: 0-Up, 1-Down [0]"
Enter OMCI edit command: 24
"POTS1 Dial Number [1111]" : 2012000984
Enter OMCI edit command: 25
"POTS1 User Name [1111]" : 2012000984
Enter OMCI edit command: 26
"POTS1 Password [1111]": password
...
Enter OMCI edit command: s
SPECIFIC profile has been saved

```

2 Make sure every variable has an assigned value.

To view the current settings of configuration variables on ONU 4/4/1 enter **gpononu profile show vars 4/4/1** command.

Activating the ONT

Activate the ONT to add it to the system. If users are adding multiple services, users would range the ONT after all the services have been added. an activated ONT is an ONT had assigned a serial number on, and the ONT port admin status is up.



Note: Only run the **gpononu set** command once to add the ONT. If the ONT has been activated and the OMCI profiles are configured for other service, users may add other bridges without resetting the ONT. If users change OMCI profiles users will need to resync/reboot the ONT. To resync ONT use the **gpononu resync <slot>[/<olt>[/<onu>]]** command. To reboot ONT use the **gpononu reboot <slot>[/<olt>[/<onu>]]** command.

- 1 To activate an ONT first run the **gpononu show** command to display the ONTs currently on the OLT, and discover the serial numbers of the ONTs.

The **gpononu show** command has options to select by slot and OLT. If users run the command without defining the slot/OLT the command will check for ONTs on every port of every card and depending on the number of cards, may take a long time to complete.

```
zSH> gpononu show 4/4
Processing list of 128
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
Free ONUs for slot 4 olt 4:
   1    2    3    4    5    6    7    8    9   10   11   12
  13   14   15   16   17   18   19   20   21   22   23   24
  25   26   27   28   29   30   31   32   33   34   35   36
  37   38   39   40   41   42   43   44   45   46   47   48
  49   50   51   52   53   54   55   56   57   58   59   60
  61   62   63   64
Discovered serial numbers for slot 4 olt 4:
sernoID  Vendor  Serial Number      Time Discovered
   1      CIGG   138543368          JUL 10 01:11:18 2014
```

- 2 Run the **gpononu set slot/OLTport/ONUport sernoID** command to associate an ONU port ID to the discovered ONT's serial number.

```
zSH> gpononu set 4/4/1 1
Onu 1 successfully enabled with serial number CIGG
138543368
```

- 3 Run the **gpononu show** command to verify the ONT is enabled, and OMCI support is added into the ONT (the associated ME profile and Generic profile can be displayed).

```
zSH> gpononu show 4/4/1
```

ONU	Name	Enabled	Model #	Serial Number	OMCI files and profiles
1	1-4-4-1	Yes	2510	CIGG 138543368	ME 2510-config1 GEN 2510-service-plan1

- 4 Run the **gpononu status** command to verify the OMCI Config State is active.

```
zSH> gpononu status 4/1/1
```

ID	Onu	OperStatus	ConfigState	Download State	OLT Rx Power	ONT Rx Power	Distance (KM)	GPON OnuStatus
1	1-4-1-1	Up	Active	NoUpgrade	-19.2 dBm	-20.0 dBm	18	Active

- Run the **port show** command to verify the ONT port admin status is up.

```
zSH> port show 1-4-1-1/gpononu
Interface 1-4-1-1/gpononu
    Administrative status: up
```

- Run the **bridge show** command to view the MAC address of the connected PC.

```
zSH> bridge show
Orig
Type VLAN/SLAN VLAN/SLAN Physical Bridge St Table
Data
-----
upl      Tagged 100 1/a/1/5/0/eth ethernet5-100/
bridge   UP S VLAN 100 default
dwn      Tagged 100 1/4/4/1/gpononu 1-4-4-501-gponport-100/bridge
UP D 00:00:86:43:3c:e4 MAC of PC
```

Testing the data bridge

Verify that the user can get data on the PC:

- Connect an ONT downlink ethernet port to a PC.

Make sure the ONT model matches the one users assigned with the Smart OMCI web tool. This example connects a ZNID-GPON-2510 to the PC.

And also make sure the ONT downlink ethernet port number matches the one users assigned with the Smart OMCI web tool for data service. In this example, users can connect either ETH 1 or ETH 2 to the PC.

- Open a command prompt on the PC and enter **ipconfig** to verify that users can get an IP address from DHCP server for the PC.
- Open an internet browser on the PC, users should be able to access the internet now.

Create uplink and downlink bridges on GPON OMCI for video

Video bridging is similar to data bridging and uses downlink/uplink bridges, however, the GPON traffic profile, GEM ports and VLANs are different.

Creating GPON traffic profile

Add the GPON traffic profile.

The following GPON traffic profile is recommended for video:

```

zSH> new gpon-traffic-profile 2
gpon-traffic-profile 2
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}: 512
traffic-class: -----> {ubr}: cbr
compensated: -----> {false}:
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: ----> {0}:
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```

Creating an uplink and downlink bridge

Create an uplink and downlink bridge for VLAN 999:

1 Create an uplink bridge interface

a Create the uplink bridge interface

The following example creates a video uplink bridge interface and enables IGMP proxy reporting with IGMP snooping.

```

zSH> bridge add 1-a-5-0/eth uplink vlan 999 tagged
igmpproxy

```

```

Adding bridge on 1-a-5-0/eth
Created bridge-interface-record 1-a-5-0-eth-999/
bridge
Bridge-path added successfully

```

b Modify the **bridge-path** for the uplink if necessary.

The following example modifies the bridge path with 30 seconds IGMP query interval.

```

zSH> bridge-path modify 1-a-5-0-eth-999/bridge vlan
999 default igmptimer 30

```

2 Create downlink bridge interface.

Create a downlink bridge on a GEM port with VLAN ID and GPON traffic profile.

Users can also specify option **video m/n**. *m* indicates the multicast control list, *n* indicates the maximum video streams. By specifying **video 0/4** in this example users can enable subscriptions up to four video streams on the interface without control list checking.

If users want to have multicast control list checking, use the **new mcast-control-entry** command to create a multicast control list first.

```
zSH> bridge add 1-4-4-901/gponport gtp 2 downlink vlan
999 tagged video 0/4
Adding bridge on 1-1-7-901/gponport
Created bridge-interface-record
1-1-7-901-gponport-999/bridge
```

- 3 Enter the **bridge show** command to view the MAC address of the connected PC.

```
zSH> bridge show
Orig
Type VLAN/SLAN VLAN/SLAN Physical Bridge St Table
Data
-----
upl Tagged 100 1/a/5/0/eth ethernet5-100/bridge UP S VLAN
100 default
dwn Tagged 100 1/4/4/1/gpononu 1-4-4-501-gponport-100/bridge UP D
00:00:86:43:3c:e4
upl Tagged 999 1/a/5/0/eth ethernet5-999/bridge UP S VLAN
999 default
dwn Tagged 999 1/4/4/1/gpononu 1-4-4-901-gponport-999/bridge UP D
00:00:87:44:0c:e7 MAC of PC
01:00:5e:0a:0a:0a D
```

Because Specific profile was already created on this ONT when configuring the data application, users do not need to create a Specific profile again.

Since users only add the ONT once, users would normally run the **gpononu set** command after have added all the services. Users may add service after activating the ONT, however if users change the OMCI profiles later, users need to resync or reboot the ONT. See the [Step 1 Activate the ONT](#) in the data application for the command and greater detail on the operation.

Testing the IPTV bridge

When using a PC and software to emulate a set top box (STB), use **ping** to verify that the video server is alive.

- 1 Connect an ONT downlink ethernet port to the customer video equipment. This example connects to a PC that runs a STB emulation software.

Make sure the ethernet port number matches the one users assigned with the Smart OMCI web tool for video service. In this example users can connect either ETH 3 or ETH 4 to the PC.
- 2 Open a command prompt on the PC and enter **ipconfig** to verify that users can get an IP address for the PC.
- 3 Ping the video server
 - a Open a DOS window
 - b Ping the upstream gateway (provided in your environment setup)

- 4 Open the STB emulation software and connect to the video server.

As long as users can ping that means there are a data path through the zNID and the MXK to the video server. Users should be able to connect to the video stream with the STB emulation software.

Create uplink and downlink bridge on GPON OMCI for VoIP

Creating GPON traffic profile

Add the GPON traffic profile.

The following GPON traffic profile is recommended for up to four VoIP phones or four POTS ports.

```
zSH> new gpon-traffic-profile 3

gpon-traffic-profile 3
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}: 512
traffic-class: -----> {ubr}: cbr
compensated: -----> {false}:
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: ----> {0}:
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved..
```

Creating an uplink and downlink bridge

Create an uplink and downlink bridge for VLAN 300:

- 1 Create the uplink bridge interface.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 300 tagged
```

```
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-300/bridge
Bridge-path added successfully
```

- 2 Create downlink bridge interface.

Create a downlink bridge on a GEM port with VLAN ID and GPON traffic profile.

```
zSH> bridge add 1-4-4-701/gponport gtp 3 downlink vlan
300 tagged
```

```
Adding bridge on 1-4-4-701/gponport
Created bridge-interface-record
1-4-4-701-gponport-300/bridge
Bridge-path added successfully
```

- 3 On MXK, run the **bridge show** command to view the MAC address of the connected VoIP phone.

```
zSH> bridge show
Orig
Type VLAN/SLAN VLAN/SLAN Physical Bridge St Table
Data
-----
upl Tagged 100 1/a/5/0/eth ethernet5-100/bridge UP S
VLAN 100 default
dwn Tagged 100 1/4/4/1/gpononu 1-4-4-501-gponport-100/bridge UP D
00:00:86:43:3c:e4
upl Tagged 999 1/a/5/0/eth ethernet5-999/bridge UP S
VLAN 999 default
dwn Tagged 999 1/4/4/1/gpononu 1-4-4-901-gponport-999/bridge UP D
00:00:87:44:0c:e7
D
01:00:5e:0a:0a:0a
dwn Tagged 300 1/4/4/1/gpononu 1-4-4-701-gponport-300/bridge UP D
00:19:c7:02:9c:6b MAC of Phone
upl Tagged 300 1/a/5/0/eth ethernet5-300/bridge UP D
00:00:86:43:3c:e4
D
00:00:86:43:ec:69
D
00:01:47:1a:e4:74
D
00:03:e3:97:bb:00
D
00:50:04:78:56:85
D
00:50:04:bf:63:3e
```

Because a Specific profile is already created on this ONT when configuring data application, users do not need to create a Specific profile again.

Since users only need to add the ONT once, users would normally run the **gpononu set** command after added all the services. Users may add service after activating the ONT, however if users change the OMCI profiles later, users need to resync or reboot the ONT. See the [Step 1 Activate the ONT](#) in the data application for the command and greater detail on the operation.

Verifying the Voice Configuration in the Generic Profile

The OMCI Transmit Gain, Receive Gain, and Fax Mode parameters can be set using the Smart OMCI Configuration Utility tool on the Zhone website (<http://www.zhone.com/support/tools/omci>) or by using the **gpononu profile update gen** command.

The **Rx Gain** and **Tx Gain** parameters configure the sensitivity of POTS ports for gain and attenuation. The Fax mode parameter defines the G.711 or T.38:

- Rx Gain
Specifies the gain value for the received signal in the form of a 2s complement number. Valid values are -120 (-12.0 dB) to 60 (+6.0 dB). The default value is 0.
- Tx Gain
Specifies the gain value for the transmit signal in the form of a 2s complement number. Valid values are -120 (-12.0 dB) to 60 (+6.0 dB). The default value is 0.
- Fax mode.
0 Passthru
1 T.38
The default value is 0 (Passthru).

Assign values to these parameters in the Generic profile. Use the **gpononu profile update gen** command, then enter the corresponding variable indexes and values.

The following example shows how to view these parameters.

View the voice configuration variables in the Generic profile with the **gpononu profile update gen** command.

```
zSH> gpononu profile update gen 2510-tripleplay-gen
Generic Profile: 2510-tripleplay-gen
1  "ETH1 Auto Detection [0]"
2  "ETH 1 Data VLAN 1 (VID or COS,VID) [0,100]" 100
3  "ETH 1 Data VLAN 2 (VID or COS,VID) [0,100]" 100
4  "ETH2 Auto Detection [0]"
5  "ETH 2 Data VLAN 1 (VID or COS,VID) [0,100]"
6  "ETH 2 Data VLAN 2 (VID or COS,VID) [0,100]"
7  "ETH3 Auto Detection [0]"
8  "ETH 3 Video VLAN 1 (VID or COS,VID) [4,999]"
9  "ETH 3 Video VLAN 2 (VID or COS,VID) [4,999]"
10 "ETH4 Auto Detection [0]"
11 "Voice VLAN [7,200]"
12 "VOIP Host IP Option: 2-static, 3-DHCP [2]"
13 "VOIP Host IP [0.0.0.0]"
14 "VOIP Netmask [0.0.0.0]"
15 "VOIP Gateway [0.0.0.0]"
16 "VOIP Server IP [0.0.0.0]"
17 "VOIP Primary DNS [0.0.0.0]"
18 "VOIP Secondary DNS [0.0.0.0]"
19 "Country Code [ 1]"
20 "Rx Gain [12]"
21 "Tx Gain [12]"
22 "Out-of-band DTMF [0]"
23 "Echo Cancel: 1-enable, 0-disable [1]"
```

```

24 "POTS1 Dial Number [1111]"
25 "POTS1 User Name [11111]"
26 "POTS1 Password [11111]"
27 "POTS2 Dial Number [2222]"
28 "POTS2 User Name [22222]"
29 "POTS2 Password [22222]"
30 "Fax Mode [1]"
31 "CID Features [63]"
32 "Call Waiting Features [3]"
33 "Call Progress or Transfer Features [255]"
34 "Call Present Features [15]"
35 "ETH 1 Admin Status: 0-Up, 1-Down [0]"
36 "ETH 2 Admin Status: 0-Up, 1-Down [0]"
37 "ETH 3 Admin Status: 0-Up, 1-Down [0]"
38 "ETH 4 Admin Status: 0-Up, 1-Down [0]"
39 "POTS 1 Admin Status: 0-Up, 1-Down [0]"
40 "POTS 2 Admin Status: 0-Up, 1-Down [0]"
Enter OMCI edit command or [s]ave, [q]uit, [h]elp:1
ID    Generic Profile: 2510-tripleplay-gen
====
=====
=====
1  Name      : $autoDetectConfigEth1
   Comment   : ETH1 Auto Detection
   Type      : string(32)
   Gen Value :
   Default Value: 0
2  Name      : $vlEth1V1
   Comment   : ETH 1 Data VLAN 1 (VID or COS,VID)
   Type      : string(32)
   Gen Value : 100
   Default Value: 0,100
3  Name      : $autoDetectConfigEth2
   Comment   : ETH2 Auto Detection
   Type      : string(32)
   Gen Value : 100
   Default Value: 0,100
<SPACE> for next page, <CR> for next line, A for all, Q
to quitq

```

Testing the VoIP configuration

- 1 Connect an ONT downlink POTS port to a VoIP phone.

Make sure the POTS port number matches the one users assigned for voice service with the Smart OMCI web tool. In this example, users can connect either POTS 1 or POTS 2 to the phone.

- 2 Pick up the phone, users should be able to hear the dial tone and be able to make and receive a phone call.

Delete the OMCI profile

This section describes how to delete the ME profile, Generic profile and Specific profile.

- The Specific profile can be deleted when the associated ONU is either activated or not activated.

Note that without the Specific profile, the OMCI provisioning on the associated ONU will be disabled.

- The ME profile and Generic profile can be deleted when they are not being used. Otherwise, an error message will be displayed stating this profile is being used.
 - The ME profile used could have a Generic profile and/or Specific profile associated with it. In that case, remove the related Generic and/or Specific profile first, and then delete the ME profile.
 - The Generic profile used could have a Specific profile associated with it. In that case, remove the related Specific profile, then delete the Generic profile.
 - An ONU is associated with this ME profile or Generic profile. In that case, remove the ME profile or Generic profile references from the ONU, then delete the ME profile or Generic profile.

Two different commands are provided to remove the ME/Generic profile references from ONUs:

gpononu set noomci command (for ONUs that haven't assigned serial numbers)

gpononu clear omci command (for ONUs that had assigned serial numbers)



Note: In the **gpononu set** command and **gpononu clear** command, the Slot ID, OLT ID, and ONU ID maybe replaced with brackets containing numbers in comma-separated series (e.g [1,4]) and in dash-separated ranges (e.g [1, 3-4]). In addition, not specifying ONU ID causes all ONUs on that OLT to be changed, and not specifying OLT ID causes all ONUs on all OLTs on that slot to be changed.

Deleting the OMCI profile when the ONU has no serial number

This section describes how to delete the ME, Generic, Specific profile when the associated ONU has no serial number on it.

This example assumes the ME profile *2510-tripleplay-me* has one Generic profile, *2510-tripleplay-gen*, and one Specific profile, *13/1/1*, associated with it:

- 1 Verify ONU 13/1/1 is not active, and the ME profile 2510-tripleplay-me and Generic profile 2510-tripleplay-gen are associated with ONU 13/1/1.

```
zSH> gpononu show 13/1/1
```

```
Slot 13 olt 1
```

ONU	Name	Enabled	Model #	Serial Number	OMCI files and profiles
1	1-13-1-1	No			ME 2510-tripleplay-me GEN 2510-tripleplay-gen

- 2 Delete the ME profile and Generic profile references to the Specific profile, and delete the Specific profile created on ONU 13/1/1 with the **gpononu set noomci** command.

This command does not change the state of the existing ONU.

Note that this command should not be used under the following two conditions:

- if the OMCI profiles were not previously set.
- if the specified ONU is currently active.

```
zSH> gpononu set 13/1/1 noomci
```

- 3 Verify the ME profile name and Generic profile name are removed from ONU 13/1/1.

```
zSH> gpononu show 13/1/1
```

```
Slot 13 olt 1
```

ONU	Name	Enabled	Model #	Serial Number	OMCI files and profiles
1	1-13-1-1	No			(none)

- 4 Verify the Specific profile on ONU 13/1/1 is also removed.

```
zSH> gpononu profile show spec 13/1/1
```

The outputs does not show 13/1/1 indicating a Specific profile created on 13/1/1 does not exist.

- 5 Delete the Generic profile, then delete the ME profile.

```
zSH> gpononu profile delete gen 2510-tripleplay-gen
```

Profile has been deleted!

```
zSH> gpononu profile delete me 2510-tripleplay-me
```

Profile has been deleted!

Deleting the OMCI profile when the ONU has serial number

This section describes how to delete a Specific profile, Generic profile and ME profile on an ONU that has serial number on it.

The following examples assume ME profile *2510-tripleplay-me* has one Generic profile, *2510-tripleplay-gen*, and one Specific profile, *13/1/1*, associated with it:

- 1** Delete a Specific profile that is used by an activated ONU. The OMCI configuration state on this ONU is changed after deleting Specific profile.
 - a** Verify the Specific profile associated ONU has serial number. And the *OmciConfigState* is Done.

```
zSH> gpononu show 13/1/1
```

```
Slot 13 olt 1
```

ONU	Name	Enabled	Model #	Serial Number	OMCI files and profiles
1	1-13-1-1	Yes	2510	ZNTS 1306	ME 2510-tripleplay-me GEN 2510-tripleplay-gen

```
zSH> gpononu status 13/1/1
```

ID	Onu	OperStatus	ConfigState	Download State	OLT Rx Power	ONT Rx Power	Distance (KM)	GPON OnuStatus
1	1-13-1-1	Up	Active	NoUpgrade	-19.2 dBm	-20.0 dBm		
18	Active							

- b** Delete the Specific profile created on this ONU.

```
zSH> gpononu profile delete spec 13/1/2
```

```
Profile has been deleted!
```

- c** Without the Specific profile, the OMCI provisioning on the associated ONU will be disabled. Verify the *OmciConfigState* is not Done.

```
zSH> gpononu status 13/1/1
```

ID	Onu	OperStatus	ConfigState	Download State	OLT Rx Power	ONT Rx Power	Distance (KM)	GPON OnuStatus
1	1-13-1-1	Up	Configuring	NoUpgrade	-19.2 dBm	-20.0 dBm		
18	Active							

- 2** Delete an ME profile and a Generic profile that is used by an ONU has serial number.

- a** Deleting an ME profile and Generic profile that is used by an ONU causes an error message to appear.

```
zSH> gpononu profile delete me 2510-tripleplay-me
```

```
ERROR! Cannot delete, profile is being used
```

```
zSH> gpononu profile delete gen 2510-tripleplay-gen
ERROR! Cannot delete, profile is being used
```

- b** Clear the serial number of the ONU, delete the ME profile and Generic profile references, and the Specific profile (if any) and disable the ONU with the **gpononu clear omci** command:

```
zSH> gpononu clear 13/1/1 omci
```

Verify the ME profile name and Generic profile name are removed from ONU 13/1/1, and the ONU is disabled.

```
zSH> gpononu show 13/1/1
Slot 13 olt 1
```

ONU	Name	Enabled	Model #	Serial Number	OMCI files and profiles
1	1-13-1-1	No		(none)	

Verify the Specific profile on ONU 13/1/1 is also removed, enter.

```
zSH> gpononu profile show spec 13/1/1
```

The outputs didn't show 13/1/1 indicating that the Specific profile does not exist on 13/1/1.

- c** Delete Generic profile, then delete ME profile.

```
zSH> gpononu profile delete gen 2510-tripleplay-gen
```

Profile has been deleted!

```
zSH> gpononu profile delete me 2510-tripleplay-me
```

Profile has been deleted!

Import and export the OMCI profile

Importing the OMCI profile

The OMCI profile import feature allows the contents of an existing OMCI profile to be overwritten with a new OMCI profile file.

Any changes in the OMCI profile file, such as adding or deleting OMCI commands in ME profile file, will cause the variables in the OMCI profile to be added, deleted, or remain the same. After importing the OMCI profile file to the existing OMCI profile, the system will reconcile the associated Generic profile and Specific profile. The user can update the variables in the Generic and Specific profile as needed.

To import a new OMCI (ME, Generic, or Specific) profile file to an existing OMCI profile, use the following commands:

- **gpononu profile import me meProfileName fileName** command

- **gpononu profile import gen** *genProfileName fileName* command
- **gpononu profile import spec** *slot/olt/onu fileName* command

The following example shows how to import an ME profile file and related configuration:

- 1 View the existing ME profiles.

```
zSH> gpononu profile show me
me1
me2
.....
```

- 2 Find the ONUs that use the selected ME profile.

```
zSH> gpononu profile find me me1
This command may take several minutes to complete.
Do you want to continue? <yes or no> [no] yes

13/1/1
```

The above example shows only ONU 13/1/1 uses *me1*.

- 3 Import the new ME profile file to the ME profile *me1*.

Importing the new ME profile file overwrites the current contents in the *me1*, and a warning message appears.

```
zSH> gpononu profile import me me1 /me/2510-mev2.txt
Profile imported.
```

Variables may have been added, deleted, or changed in the ME Profile "me1". The Generic and Specific profiles associated with the ME profile "me1" have been reconciled to include these variable modifications (if any). Please edit the relevant Generic and Specific profiles accordingly.

- 4 Find the relevant Generic profile, and then specify the desired values to the variables in the Generic profile.

```
zSH> gpononu show 13/1/1
```

```
Slot 13 olt 1
```

ONU	Name	Enabled	Model #	Serial Number	OMCI files and profiles
1	1-13-1-1	Yes	2510	ZNTS 1306	ME 2510-tripleplay-me GEN 2510-tripleplay-gen

The above example shows ONU 13/1/1 uses *2510-tripleplay-gen*. Then update the generic profile *2510-tripleplay-gen* as desired.

```
zSH> gpononu profile update gen 2510-tripleplay-gen
Generic Profile: 2510-tripleplay-gen
```

```

 1 "newvariable"                                the new variable
 2 "ETH1 Auto Detection [1]"
 3 "ETH 1 Data VLAN 1 (VID or COS,VID) [0,100]"
 4 "ETH2 Auto Detection [0]"
 5 "ETH 2 Data VLAN 1 (VID or COS,VID) [0,100]"
 6 "ETH3 Auto Detection [1]"
 7 "ETH 3 Video VLAN 1 (VID or COS,VID) [4,999]"
 8 "ETH4 Auto Detection [0]"
 9 "Voice VLAN [7,200]"
10 "VOIP Host IP Option: 2-static, 3-DHCP [2]"
11 "VOIP Host IP [0.0.0.0]"
12 "VOIP Netmask [0.0.0.0]"
13 "VOIP Gateway [0.0.0.0]"
14 "VOIP Server IP [0.0.0.0]"
15 "VOIP Primary DNS [0.0.0.0]"
16 "VOIP Secondary DNS [0.0.0.0]"
17 "Country Code [ 1]"
18 "Rx Gain [0]"
19 "Tx Gain [0]"
20 "Out-of-band DTMF [0]"
21 "Echo Cancel: 1-enable, 0-disable [1]"
22 "POTS1 Dial Number [1111]"
23 "POTS1 User Name [11111]"
24 "POTS1 Password [11111]"
25 "POTS2 Dial Number [2222]"
26 "POTS2 User Name [22222]"
27 "POTS2 Password [22222]"
28 "Fax Mode [0]"
29 "CID Features [63]"
30 "Call Waiting Features [3]"
31 "Call Progress or Transfer Features [255]"
32 "Call Present Features [15]"
33 "ETH 1 Admin Status: 0-Up, 1-Down [0]"
34 "ETH 2 Admin Status: 0-Up, 1-Down [0]"
35 "ETH 3 Admin Status: 0-Up, 1-Down [0]"
36 "ETH 4 Admin Status: 0-Up, 1-Down [0]"
37 "POTS 1 Admin Status: 0-Up, 1-Down [0]"
38 "POTS 2 Admin Status: 0-Up, 1-Down [0]"
Enter OMCI edit command: 1
Enter value: ###
...
Enter OMCI edit command: s
GENERIC profile has been saved
```

5 Specify the desired values to the variables in the relevant Specific profile.

```
zSH> gpononu profile update spec 13/1/1
Specific Profile: 13/1/1
 1 "newvariable"                                the new variable
```

```

2 "ETH1 Auto Detection [1]"
3 "ETH 1 Data VLAN 1 (VID or COS,VID) [0,100]"
4 "ETH2 Auto Detection [0]"
5 "ETH 2 Data VLAN 1 (VID or COS,VID) [0,100]"
6 "ETH3 Auto Detection [1]"
7 "ETH 3 Video VLAN 1 (VID or COS,VID) [4,999]"
8 "ETH4 Auto Detection [0]"
9 "Voice VLAN [7,200]"
10 "VOIP Host IP Option: 2-static, 3-DHCP [2]"
11 "VOIP Host IP [0.0.0.0]"
12 "VOIP Netmask [0.0.0.0]"
13 "VOIP Gateway [0.0.0.0]"
14 "VOIP Server IP [0.0.0.0]"
15 "VOIP Primary DNS [0.0.0.0]"
16 "VOIP Secondary DNS [0.0.0.0]"
17 "Country Code [ 1]"
18 "Rx Gain [0]"
19 "Tx Gain [0]"
20 "Out-of-band DTMF [0]"
21 "Echo Cancel: 1-enable, 0-disable [1]"
22 "POTS1 Dial Number [1111]"
23 "POTS1 User Name [1111]"
24 "POTS1 Password [1111]"
25 "POTS2 Dial Number [2222]"
26 "POTS2 User Name [2222]"
27 "POTS2 Password [2222]"
28 "Fax Mode [0]"
29 "CID Features [63]"
30 "Call Waiting Features [3]"
31 "Call Progress or Transfer Features [255]"
32 "Call Present Features [15]"
33 "ETH 1 Admin Status: 0-Up, 1-Down [0]"
34 "ETH 2 Admin Status: 0-Up, 1-Down [0]"
35 "ETH 3 Admin Status: 0-Up, 1-Down [0]"
36 "ETH 4 Admin Status: 0-Up, 1-Down [0]"
37 "POTS 1 Admin Status: 0-Up, 1-Down [0]"
38 "POTS 2 Admin Status: 0-Up, 1-Down [0]"
Enter OMCI edit command: 1
Enter value: ###
...
Enter OMCI edit command: s
SPECIFIC profile has been saved

```

6 Send the latest OMCI commands from OLT to ONU.

```
zSH> gpononu resync 13/1/1
```

Exporting the OMCI profile file

The OMCI profile export feature allows user to copy the contents of an OMCI profile to a new OMCI profile file, and store this file into the uplink compact flash. Later this OMCI profile file could be shared with other devices.

To export the contents of an OMCI (ME, Generic, or Specific) profile to a new OMCI profile file, use the following commands:

- **gpononu profile export me** *meProfileName* *fileName* command
- **gpononu profile export gen** *genProfileName* *fileName* command
- **gpononu profile export spec** *slot/olt/onu* *fileName* command

The following example copies an ME profile file from an MXK to another MXK:

- 1 On MXK A, copy the contents of ME profile *me1* to a new ME profile file, name it as *2510-mev1.txt*, and save it to *root/pub* directory.

```
zSH> cd pub
```

```
zSH> gpononu profile export me me1 2510-mev1.txt
Profile exported.
```

By default, the MXK runs as a TFTP server enabling files stored in the *root/pub* folder to be downloaded to other devices with connectivity to the MXK.

- 2 On MXK B, download the ME profile file *2510-mev1.txt* from the MXK A (IP address 172.42.15.19) to the local directory *me*, and name it as *2510-me.txt*.

```
zSH> file download 172.42.15.19 /pub/2510-mev1.txt /
me/2510-me.txt
File download successful
```

- 3 Import the ME profile file *2510-me.txt* to an existing ME profile *2510me*.

```
zSH> gpononu profile import me 2510me /me/2510-me.txt
Profile imported.
```

Variables may have been added, deleted, or changed in the ME Profile "me1". The Generic and Specific profiles associated with the ME profile "me1" have been reconciled to include these variable modifications (if any). Please edit the relevant Generic and Specific profiles accordingly.

- 4 Update the Generic profile or Specific profile as desired.

See steps 4 to 6 in [Importing the OMCI profile on page 720](#).

Unified Service Provisioning GPON zNID installation

Building on Dynamic OMCI to provide automatic software downloads, pre-provisioning support, and zNID configuration “master” in MXK — Unified Service Provisioning (USP) provides a single provisioning method for Zhone GPON zNIDs via the MXK and ZMS.

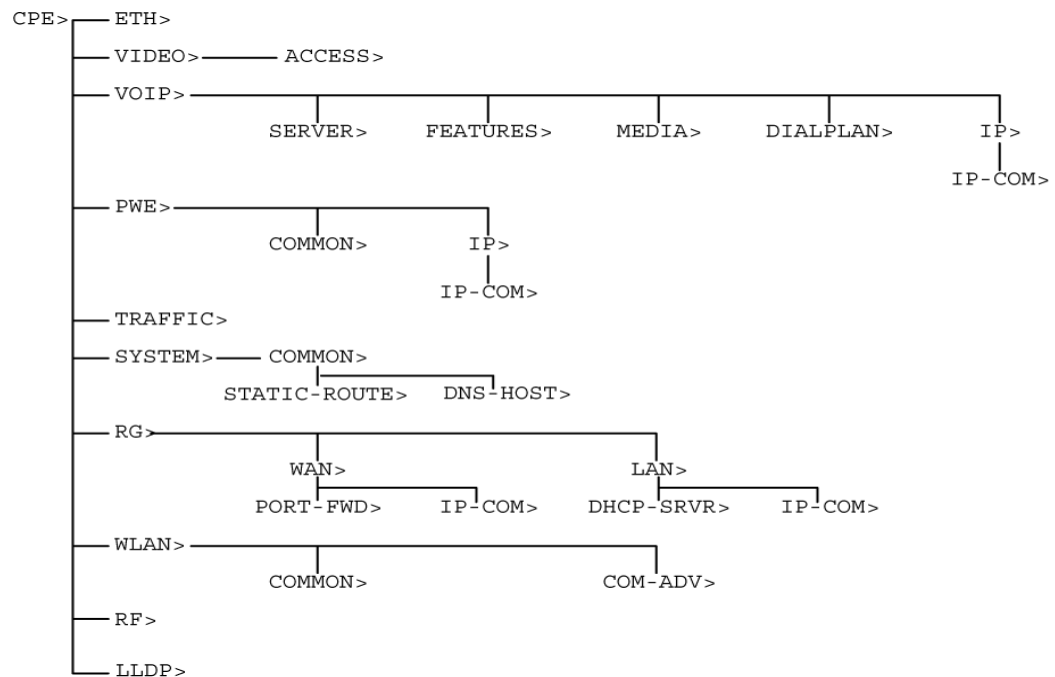
Unified Service Provisioning (USP) accesses all the management interfaces, now extending to work with features that were only accessible through SNMP or the CPE device’s Residential Gateway web based user interface.

Unified Service Provisioning section includes the following topics:

- [CPE Menu System, page 725](#)
- [One GEM port Allocated for Internal Communication with the ONT for USP, page 728](#)
- [GEM Ports Assignments in USP, page 729](#)
- [GPON Traffic Profile Assignment in USP, page 731](#)
- [Dynamic OMCI GPON zNID Installation, page 734](#)
- [Residential Gateway \(RG\) Features Provisioning, page 791](#)
- [AutoConfiguration and AutoDiscovery OMCI GPON zNID Installation, page 861](#)
- [VoIP Phone with LLDP-MED Network Policy, page 880](#)
- [Additional Features in Unified Service Provisioning with “bridge add” Command, page 885](#)
- [Post Configuration in USP, page 898](#)

CPE Menu System

A CPE menu system as shown in [Figure 112](#) is implemented in Unified Service Provisioning.

Figure 112: CPE menu system in Unified Service Provisioning

Using the CPE Command Shell for Unified Service Provisioning

To provision and manage CPE service on ONUs, users can either type a single-line CPE profile macro command, or use the CPE command shell.

The following examples show how to use the CLI command menu shell on CPE profiles.

- 1 To enter a command shell:

```

zSH> ↵
zSH> cpe↵
zSH> CPE> voip↵
zSH> CPE > VOIP> server↵
zSH> CPE > VOIP> SERVER>
Or

```

```

zSH> cpe voip server↵
zSH> CPE > VOIP> SERVER>

```

- 2 To get supported commands in the current command shell:

```

zSH> CPE> VOIP> SERVER> ?↵
Error: Invalid argument "?"
cpe voip server <add | delete | find | modify | show>

```

- 3 To get help on the supported commands in the current command shell:

```
zSH> CPE> VOIP> FEATURES> help←
```

```
cpe voip features show < index | profile-name | all >
    Display single or multiple profiles
```

```
cpe voip features add <profile-name>
    [ announcement-type < silence | reordertone | fastbusy | voice | na > ]
        specifies the treatment when a subscriber goes off hook but does
        not attempt a call
    [ cid-features < calling-number | calling-name | cid-block | cid-number
        |cid-name | anonym-block | all | none > ]
        bit map of the caller ID features
    [ call-waiting-features < call-waiting | cid-announcement | all | none
        > ]
        bit map of the call waiting features
    [ call-progress-or-transfer-features < 3-way | call-transfer | call-hold
        | call-park |do-not-disturb | flash-on-emergency |
        emergency-hold |6-way | all | none > ]
        bit map of the call processing features
    [ call-presentation-features < msg-wait-splash-ring |
        msg-wait-special-dial-tone |msg-wait-visual | call-fwd | all | none > ]
        bit map of call presentation features
```

To clear a bit map value, simply place a minus sign in from of the argument.
Example: "-calling-name" clears the calling-name value in the cid-features.

To enable all features in a bit-map use the "all" keyword.

To clear all features in a bit-map use the "none" keyword.

```
[ hotline < disabled | hot | warm > ]
    When the hotline is hot, the phone will immediately dial the
    hotline number.
    When the hotline is warm, the phone wait for the period specified
    in warmline-timer in ms before automatically dial the hotline
    number.
[ hotLine-number ] - hot line number to be given
[warmLine-Timer ]- timer to wait for the specified period before
    dialing hotLine-number
```

This command creates a new profile. The <profile-name> must be supplied and must be unique for profile type. The profile index will be automatically generated

```
cpe voip features modify < index | profile-name > [ arguments ]
    Modify a profile using the profile's <index> or <profile-name>.
    See "add" command for available [ arguments ]
```

```
cpe voip features delete < index | profile-name >
    Delete a profile using the profile's <index> or <profile-name>
```

```
cpe voip features find < index | profile-name >
    Find all cpe-voip-subscriber profiles that reference the cpe-voip-features
```

- 4 To perform Tab-completions in the current command shell:

```
zSH> CPE> VOIP> SERVER> add metaswitch-sip
primary-server 172.16.60.51 s<Tab>
secondary-server
signalling-dscp
signalling-protocol
sip-domain
sip-reg-exp-time
sip-reg-retry-time
sip-registrar
sip-rereg-head-start-time
softswitch

zSH> CPE> VOIP> SERVER> add metaswitch-sip
primary-server 172.16.60.51 se<Tab>

zSH> CPE> VOIP> SERVER> add metaswitch-sip
primary-server 172.16.60.51 secondary-server
```

- 5 To exit from the current command shell:

```
zSH> CPE> VOIP> SERVER> exit↵
zSH> CPE> VOIP>
```

Or use the short cut key **x**:

```
zSH> CPE> VOIP> SERVER> x↵
zSH> CPE> VOIP>
```

- 6 To exit from the CPE command shell:

```
zSH> CPE> VOIP> SERVER> quit↵
zSH>
```

Or use the short cut key **q**:

```
zSH> CPE> VOIP> SERVER> q↵
zSH>
```

One GEM port Allocated for Internal Communication with the ONT for USP

When using Unified Service Provisioning, a GEM port will be internally created for the SNMP interface for USP. The internally created GEM port IDs start at 3829, and grow upward. There can be up to 64 of GEM ports since there are maximum 64 ONUs per OLT link. Each of those GEM ports consumes a Reserved VLAN.

When users configuring Reserved VLANs, users need to ensure one additional GEM port to VLAN mapper is accounted for each ONU. For the details, refer to [Configurable range for Reserved VLAN per GEM port, page 951](#).

GEM Ports Assignments in USP

When using Unified Service Provisioning, any GEM port ID in the range of 257 to 3828 is allowed to be associated with any ONU *except for* GEM port ID 5xx, where xx must be the ONU port number in the range from 1 to 64.



Note: Some zNIDs models may reserve some GEM ports for different usage. Check with the *zNID Configuration Guide* to determine the available Unified Service Provisioning GEM port IDs.

For example, Zhone does not recommend users to use the GEM port ID 5xx, GEM ports in the range of 5xx are reserved for CPE managers.

Users can specify GEM ports field or skip it in the **bridge add** command:

- [Auto Assigned GEM ports, page 729](#) (i.e. auto-assigned GEM ports)
- [Arbitrary GEM ports, page 729](#) (i.e. manual-specified GEM ports)

Auto Assigned GEM ports

The GEM port ID field is not a mandatory field in the **bridge add** command. The system will automatically pick the available GEM port IDs for ONUs.

This is an example in the USP — AutoConfiguration and AutoDiscovery OMCI GPON zNID installation. In this example, users add data services to the “tripleplay” service template without specifying a GEM port.

```
zSH> bridge add tripleplay gtp 10240 downlink vlan 102 tagged rg-brouted eth [1-2]
Adding bridge on tripleplay
Created bridge-interface-record 1-0-0-257-gponport-102/bridge GEM port 257 has been assigned
automatically
CPE Connection 1-0-0-257/gponport/1/1/0/0 has been created
CPE Connection 1-0-0-257/gponport/1/2/0/0 has been created
```

Arbitrary GEM ports

When creating a GEM port with the **bridge add** command, users specify both the *ONU interface ID* and *GEM port ID*. This is an example in the USP — Dynamic OMCI GPON zNID installation:

Note that each of these GEM port IDs needs to be unique for the OLT port.

```
zSH> bridge add 1-1-3-1/gpononu gem 610 gtp 1 downlink
vlan 1001 tagged eth 1
Adding bridge on 1-1-3-1/gpononu
Created bridge-interface-record 1-1-3-610-gponport-1001/
bridge
CPE Connection 1-1-3-610/gponport/1/1/0/0 has been
created
```

If the specified GEM port ID is free, then it will be assigned to the ONU.

If the GEM port ID already exists and has been used by the same ONU, it will be reused.

If it has been assigned to a different ONU, an error message appears and the command will fail.

To view what GEM port IDs are used in the ONU, use the **gpononu gempports** command.

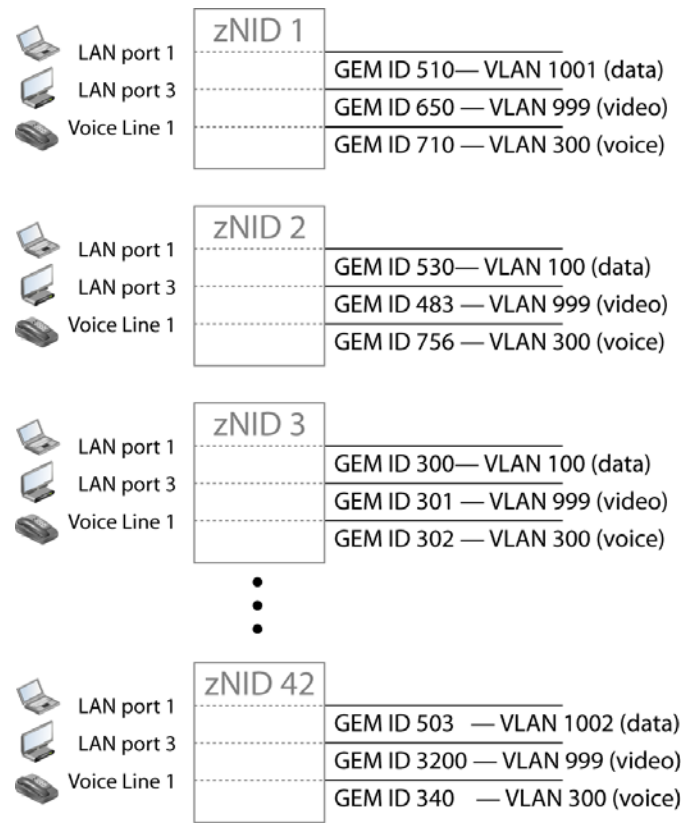
The **gpononu gempports** command has options to select by slot, OLT, or ONU. If users run the command without defining the slot/OLT/ONU, the command will check for ONTs on every port of every card and depending on the number of cards, may take a long time to complete.

zSH> onu gempports 1/3/1

ONU	GEM Port	Admin	traf prof	compn share	Fixed UBR	Fixed CBR	Assured	Max	Extra	allocId	DBA
					Bandwidth	Bandwidth	Bandwidth	Bandwidth	Bandwidth		
1-1-3-1	1-1-3-610	Up	1	False	False	2.048	0	n/a	n/a	n/a	510 n/a
	1-1-3-710	Up	3	False	False	0	0.512	n/a	n/a	n/a	641 n/a
	1-1-3-650	Up	2	False	False	0	0.512	n/a	n/a	n/a	640 n/a

The **bridge add** command example also defines which VLAN is matched to the GEM port. As shown in [Figure 113](#). Depends on your implementation, users can specify one VLAN for one service, and assign the same VLAN to different ONU GEM ports. In this example, the service provider uses zNID 1 and zNID 42 for businesses, uses zNID 2 and 3 for residential area.

Figure 113: Example GPON VLAN implementation



GPON Traffic Profile Assignment in USP

The GPON Traffic Profile field is not a mandatory field in the **bridge add** command, users can either specify the GTP field or skip it.

- [Auto Assigned GTPs, page 731](#)
- [Manual Specified GTPs, page 734](#)

Auto Assigned GTPs

If the GTP field is skipped in the **bridge add** command, the system will automatically assign a default GTP based on the following service types:

- For data services, the default GTP is 1100000000:

```
zSH> get gpon-traffic-profile 1100000000
gpon-traffic-profile 1100000000
guaranteed-upstream-bw: -> {0}
traffic-class: -----> {ubr}
compensated: -----> {false}
shared: -----> {false}
dba-enabled: -----> {true}
dba-fixed-us-ubr-bw: ----> {128}
dba-fixed-us-cbr-bw: ----> {0}
dba-assured-us-bw: -----> {0}
dba-max-us-bw: -----> {1000000}
dba-extra-us-bw-type: ---> {nonassured}
```

- For 2-port POTS services, the default GTP is 1100000001:

```
zSH> get gpon-traffic-profile 1100000001
gpon-traffic-profile 1100000001
guaranteed-upstream-bw: -> {0}
traffic-class: -----> {ubr}
compensated: -----> {false}
shared: -----> {false}
dba-enabled: -----> {true}
dba-fixed-us-ubr-bw: ----> {128}
dba-fixed-us-cbr-bw: ----> {0}
dba-assured-us-bw: -----> {256}
dba-max-us-bw: -----> {384}
dba-extra-us-bw-type: ---> {nonassured}
```

- For 4-port POTS services, the default GTP is 1100000002:

```
zSH> get gpon-traffic-profile 1100000002
gpon-traffic-profile 1100000002
guaranteed-upstream-bw: -> {0}
traffic-class: -----> {ubr}
compensated: -----> {false}
shared: -----> {false}
dba-enabled: -----> {true}
dba-fixed-us-ubr-bw: ----> {128}
```

```
dba-fixed-us-cbr-bw: ----> {0}
dba-assured-us-bw: -----> {384}
dba-max-us-bw: -----> {1000000}
dba-extra-us-bw-type: ---> {nonassured}
```

- For 8-port POTS services, the default GTP is 1100000003:

```
zSH> get gpon-traffic-profile 1100000003
gpon-traffic-profile 1100000003
guaranteed-upstream-bw: -> {0}
traffic-class: -----> {ubr}
compensated: -----> {false}
shared: -----> {false}
dba-enabled: -----> {true}
dba-fixed-us-ubr-bw: ----> {128}
dba-fixed-us-cbr-bw: ----> {0}
dba-assured-us-bw: -----> {896}
dba-max-us-bw: -----> {1000000}
dba-extra-us-bw-type: ---> {nonassured}
```

- For 24-port POTS services, the default GTP is 1100000004:

```
zSH> get gpon-traffic-profile 1100000004
gpon-traffic-profile 1100000004
guaranteed-upstream-bw: -> {0}
traffic-class: -----> {ubr}
compensated: -----> {false}
shared: -----> {false}
dba-enabled: -----> {true}
dba-fixed-us-ubr-bw: ----> {128}
dba-fixed-us-cbr-bw: ----> {0}
dba-assured-us-bw: -----> {2944}
dba-max-us-bw: -----> {1000000}
dba-extra-us-bw-type: ---> {nonassured}
```

- For 2-port PWE services, the default GTP is 1100000005:

```
zSH> get gpon-traffic-profile 1100000005
gpon-traffic-profile 1100000005
guaranteed-upstream-bw: -> {0}
traffic-class: -----> {ubr}
compensated: -----> {false}
shared: -----> {false}
dba-enabled: -----> {true}
dba-fixed-us-ubr-bw: ----> {0}
dba-fixed-us-cbr-bw: ----> {6016}
dba-assured-us-bw: -----> {0}
dba-max-us-bw: -----> {1000000}
dba-extra-us-bw-type: ---> {nonassured}
```

- For 4-port PWE services, the default GTP is 1100000006:

```
zSH> get gpon-traffic-profile 1100000006
gpon-traffic-profile 1100000006
```



```

guaranteed-upstream-bw: -> {0}
traffic-class: -----> {ubr}
compensated: -----> {false}
shared: -----> {false}
dba-enabled: -----> {true}
dba-fixed-us-ubr-bw: ----> {0}
dba-fixed-us-cbr-bw: ----> {12032}
dba-assured-us-bw: -----> {0}
dba-max-us-bw: -----> {1000000}
dba-extra-us-bw-type: ---> {nonassured}

```

- For 8-port PWE services, the default GTP is 1100000007:

```

zSH> get gpon-traffic-profile 1100000007
gpon-traffic-profile 1100000007
guaranteed-upstream-bw: -> {0}
traffic-class: -----> {ubr}
compensated: -----> {false}
shared: -----> {false}
dba-enabled: -----> {true}
dba-fixed-us-ubr-bw: ----> {0}
dba-fixed-us-cbr-bw: ----> {24000}
dba-assured-us-bw: -----> {0}
dba-max-us-bw: -----> {1000000}
dba-extra-us-bw-type: ---> {nonassured}

```

- For 24-port PWE services, the default GTP is 1100000008:

```

zSH> get gpon-traffic-profile 1100000008
gpon-traffic-profile 1100000008
guaranteed-upstream-bw: -> {0}
traffic-class: -----> {ubr}
compensated: -----> {false}
shared: -----> {false}
dba-enabled: -----> {true}
dba-fixed-us-ubr-bw: ----> {0}
dba-fixed-us-cbr-bw: ----> {72000}
dba-assured-us-bw: -----> {0}
dba-max-us-bw: -----> {1000000}
dba-extra-us-bw-type: ---> {nonassured}

```

This is an example in the AutoConfiguration and AutoDiscovery OMCI GPON zNID installation. In this example, users can add video services to the “tripleplay” service template without specifying GEM ports and GTPs.

```

zSH> bridge add tripleplay downlink vlan 203 tagged rg-bridged eth 3 video 0/10
Adding bridge on tripleplay
Created bridge-interface-record 1-0-0-258-gponport-203/bridge GEM 258 is auto-assigned
CPE Connection 1-0-0-258/gponport/12/3/0/0 has been created

```

Manual Specified GTPs

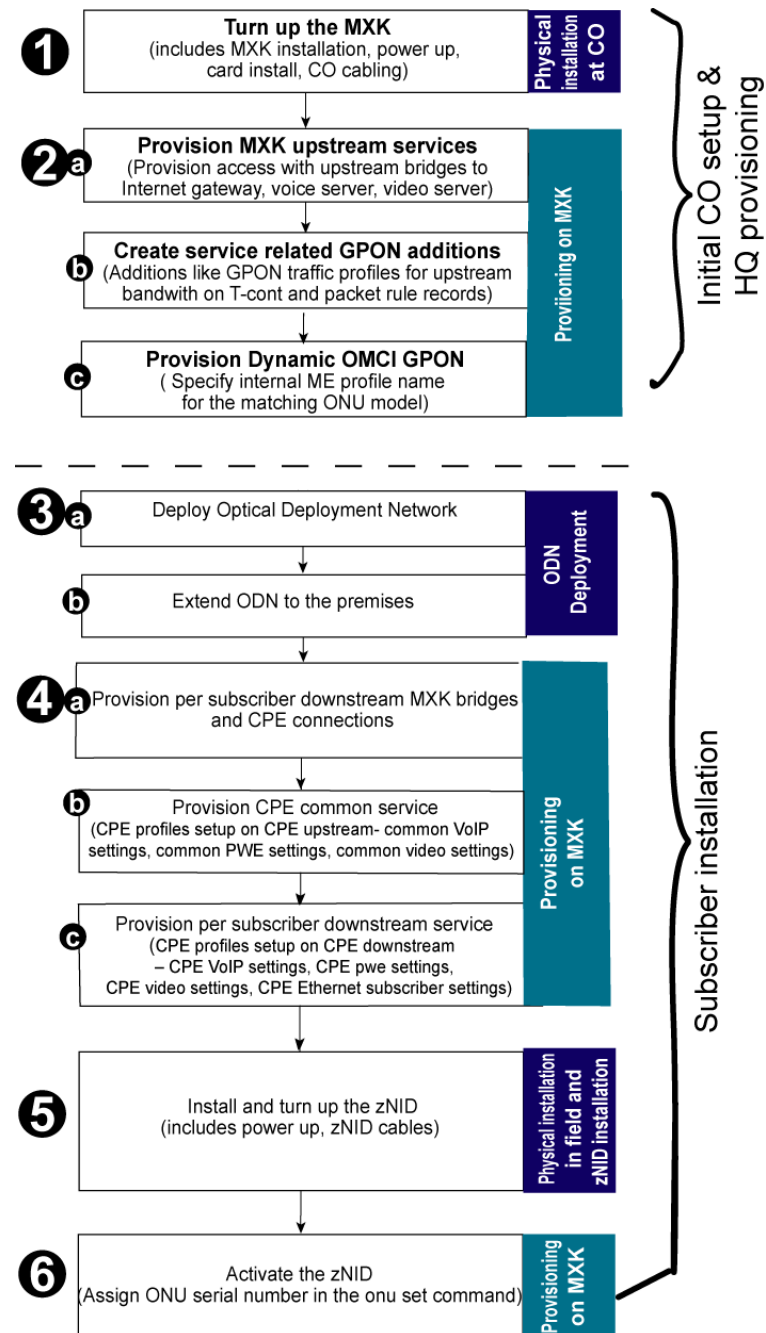
This is an example in the Residential Gateway OMCI GPON zNID installation. In this example, users create video services on a downlink bridge for ONU 4/1/1, and associate GTP 2 with GEM port 403.

```
zSH> bridge add 1-4-1-1/gpononu gem 403 gtp 2 downlink vlan 200 tagged video 0/4 eth 2
rg-bridged
Adding bridge on 1-4-1-1/gpononu
Created bridge-interface-record 1-4-1-403-gponport-200/bridge
CPE Connection 1-4-1-403/gponport/12/2/0/0 has been created
```

Dynamic OMCI GPON zNID Installation

This section provides information on how to install OMCI-based GPON zNID with Dynamic OMCI. [Figure 114](#) shows the overall flowchart of the installation procedure. This section includes the following topics:

- [Dynamic OMCI Overview, page 735](#)
- [OMCI GPON zNID Installation with Dynamic OMCI for Triple Services, page 748](#)
- [Viewing all services on an ONU, page 789](#)
- [Deletion of CPE profiles and CPE connection that associated on an ONU, page 790](#)

Figure 114: Installation procedure for OMCI GPON zNIDs with Dynamic OMCI

Dynamic OMCI Overview

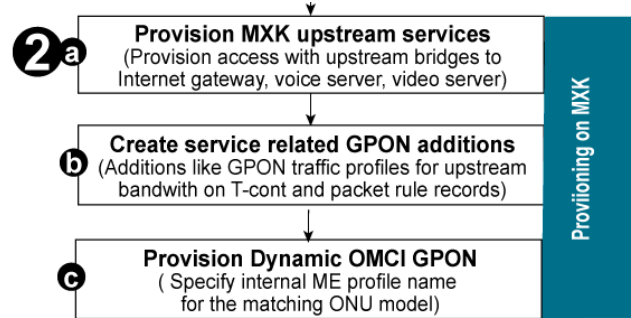
In an effort to improve the MXK user experience, Zhone has added enhanced functionality to provision CPE devices. The user is now able to use a single macro command to provision a bridge on the MXK and the attached CPE at the same time. Voice, Video, and Data services are configured on the MXK and flow through to the CPE.

This section describes the following terms used in Dynamic OMCI:

- [Internal ME Profiles, page 736](#)
- [Bridge add Commands in Dynamic OMCI, page 739](#)
- [CPE Traffic Management Profiles, page 742](#)
- [CPE Profiles, page 746](#)
- [CPE UNI Ports Default States, page 747](#)

Internal ME Profiles

Figure 115: Internal ME profile in Dynamic OMCI



Internal ME profiles are the indicator of Unified Service Provisioning.

Zhone provides internal ME profiles for supported Zhone GPON ONTs. The format of a internal ME profile name is *zhone-ZnidModel*.

As shown in the flowchart step 2c, by specifying an internal ME profile name in the initial setup (use the command **onu set OnuInterface meprofile InternalMEProfileName**) on an ONU, the MXK knows the model of this ONU, and will provision that ONU with Unified Service Provisioning. For example, internal ME profile *zhone-5114* defines there are 4 Ethernet ports, 2 POTS ports, and 4 PWE ports on the ZNID-GPON-5114, which supports both SIP and H.248 VoIP signaling.

Zhone also provides a universal ME profile for any zNIDs: *zhone-default*.

Showing supported internal ME profiles in the MXK

- 1 Users can use the **gpononu profile show internal-me** [*<partial-me-name>*] command to find valid internal ME profiles in the MXK.

```

zSH> gpononu profile show internal-me
zhone-1e          1 ETH
zhone-2301        1 GE
zhone-2311        1 GE
zhone-2402        2 GE
zhone-2403        2 GE + 1 RFV)
zhone-2424        4 GE + 2 POTS
zhone-2425        4 GE + 2 POTS + 1 RFV
zhone-2426        4 GE + 2 POTS + 1 WiFi(wlan1-wlan4) + 1 USB
  
```

zhone-2427	4 GE + 2 POTS + 1 WiFi(wlan1-wlan4) + 1 RFV + 1 USB
zhone-2501	1 GE
zhone-2504	4 GE
zhone-2510	4 FE + 2 POTS
zhone-2510a	4 FE + 2 POTS
zhone-2511	4 FE + 2 POTS + 1 RFV
zhone-2516	4 GE + 2 POTS + 2 WLAN
zhone-2517	4 GE + 2 POTS + 2 WLAN + 1 RFV
zhone-2520	4 FE + 4 POTS
zhone-2543	4 GE + 2 POTS + 1 RFV
zhone-2608t	8 GE PoE
zhone-2624a	4 GE + 2 POTS
zhone-2624p	4 GE PoE + 2 POTS
zhone-2628a	8 GE + 2 POTS
zhone-2628p	4 GE PoE(eth1-eth4) + 4 GE(eth5-eth8) + 2 POTS
zhone-2628t	8 GE PoE + 2 POTS
zhone-2644a	4 GE + 4 POTS
zhone-2644p	4 GE PoE + 4 POTS
zhone-2648a	8 GE + 4 POTS
zhone-2648p	4 GE PoE(eth1-eth4) + 4GE(eth5-eth8) + 4 POTS
zhone-2648t	8 GE PoE + 4 POTS
zhone-2802p	2 GE PoE
zhone-2804p	4 GE PoE
zhone-4220	2 GE + 2 POTS + 1 USB
zhone-4221	2 GE + 2 POTS + 1 RFV + 1 USB
zhone-4222	2 GE + 1 HCNA(eth3) + 2 POTS + 1 USB
zhone-4223	2 GE + 1 HCNA(eth3) + 2 POTS + 1 RFV + 1 USB
zhone-4224	2 GE + 1 HCNA(eth3) + 1 HPNA(eth4) + 2 POTS + 1 USB
zhone-4225	2 GE + 1 HCNA(eth3) + 1 HPNA(eth4) + 2 POTS + 1 RFV + 1 USB
zhone-4226	6 GE + 2 POTS + 1 USB
zhone-4240	2 GE + 4 POTS + 1 USB
zhone-4241	2 GE + 4 POTS + 1 RFV + 1 USB
zhone-4242	2 GE + 1 HCNA(eth3) + 4 POTS + 1 USB
zhone-4243	2 GE + 1 HCNA(eth3) + 4 POTS + 1 RFV + 1 USB
zhone-4244	2 GE + 1 HCNA(eth3) + 1 HPNA(eth4) + 4 POTS + 1 USB
zhone-4222a	2 GE + 2 POTS + 1 USB
zhone-4224a	4 GE + 2 POTS + 1 USB
zhone-4222h	2 GE + 1 HCNA(eth3) + 2 POTS + 1 USB
zhone-4224h	4 GE + 1 HCNA(eth5) + 2 POTS + 1 USB
zhone-5114	4 GE + 2 POTS + 4 T1/E1
zhone-5120	4 GE + 2 POTS + 8 T1/E1
zhone-7310	1 FE + 8 POTS + 2 T1/E1
zhone-8224	24 FE
zhone-8324	24 FE + 24 POTS
zhone-8424	24 FE
zhone-8524	24 FE + 24 POTS
zhone-9108	9 GE PoE + SFP(eth10) + 8 POTS + 1 USB
zhone-9208	9 GE PoE + SFP(eth10) + 8 POTS + 8 RF(MOCA) + 1 USB
zhone-9308	9 GE PoE + SFP(eth10) + 8 POTS + 8 RF(MOCA+RFV) + 1 USB
zhone-9440	5 GE PoE + SFP(eth10) + 4 T1/E1 + 1 USB
zhone-9444	5 GE PoE + SFP(eth10) + 4 POTS + 4 T1/E1 + 1 USB
zhone-9480	9 GE PoE + SFP(eth10) + 8 T1/E1 + 1 USB
zhone-9488	9 GE PoE + SFP(eth10) + 8 POTS + 8 T1/E1 + 1 USB
zhone-cig	24 FE + 24 POTS

zhone-default 24 FE + 24 POTS + 24 T1/E1 + 24 RFV

- 2 Users can find internal ME profiles that contain the same pattern by specifying partial ME name in the **gpononu profile show internal-me** command.

```
zSH> onu profile show internal-me zhone-25
zhone-2501                      1 GE
zhone-2504                      4 GE
zhone-2510                      4 FE + 2 POTS
zhone-2510a                    4 FE + 2 POTS
zhone-2511                      4 FE + 2 POTS + 1 RFV
zhone-2516                      4 GE + 2 POTS + 2 WLAN
zhone-2517                      4 GE + 2 POTS + 2 WLAN + 1 RFV
zhone-2520                      4 FE + 4 POTS
zhone-2543                      4 GE + 2 POTS + 1 RFV
```

Associating or removing internal ME profile from the ONU

The following two procedures show how to associate or remove internal ME profile from individual ONU or ONUs in a range.

- 1 Users can use the **onu set meprof** command to set the internal ME profiles for the individual ONT. After that, unified service provisioning supports are associated with the ONT.

zSH> **onu set 1/1/5 meprof zhone-5114**

```
zSH> onu show 1/1/5
```

ONU	Name	Enabled	Model #	Serial Number	OMCI files and profiles
5	1-1-1-5	No			ME zhone-5114

Note : NULL Model String indicates not able to get model ID

To clear the internal ME profile from this ONT, use the **onu set noomci** command.

zSH> **onu set 1/1/5 noomci**

```
zSH> onu show 1/1/5
```

ONU	Name	Enabled	Model #	Serial Number	OMCI files and profiles
5	1-1-1-5	No			(none)

Note : NULL Model String indicates not able to get model ID

- 2 Or, users can use the **onu set meprof** and **onu set noomci** command for setting and clearing ME profile from ranges of ONUs.

The Slot ID, OLT ID, and ONU ID maybe replaced with brackets containing numbers in comma-separated series (e.g. **[1,4]**), in dash-separated ranges (e.g. **[1, 3-8]**). In addition, if there is no ONU ID specified, that means all ONUs on that OLT will be changed; and if there is no OLT ID specified, that means all ONUs on all OLTs on that slot will be changed.

This example shows set/clear ME profiles for all ONUs under slot 1 and OLT port 3.

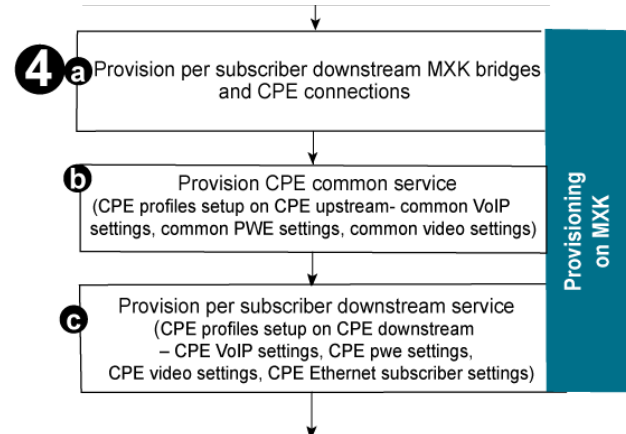
```
zSH> onu set 1/3 meprof zhone-2628p
zSH> onu set 1/3 noomci
```

This example shows set/clear ME profiles for ONU 1 to ONU 10 under slot 1 and OLT port 3.

```
zSH> onu set 1/3/[1-10] meprof zhone-2628p
zSH> onu set 1/3/[1-10] noomci
```

Bridge add Commands in Dynamic OMCI

Figure 116: Dynamic bridging in Dynamic OMCI

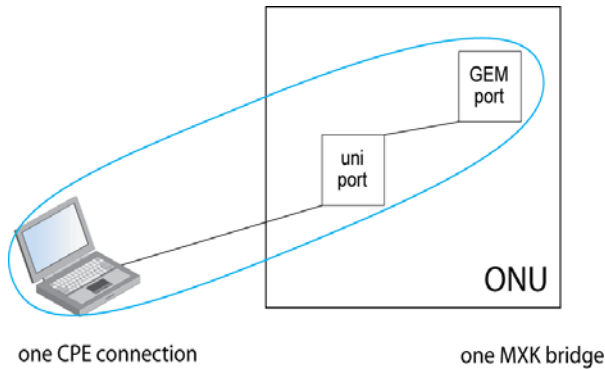


Dynamic bridging is used in step 4a in the flowchart. It uses a single macro command “**bridge add**” to create both MXK bridge and CPE connections, and define the bridge-related parameters for service types.

In Dynamic OMCI, the MXK bridge and CPE connection can have one-to-one mappings.

As shown in [Figure 117](#), the one-to-one mapping is one MXK bridge created on a GEM port that maps to one CPE connection created on an ONU UNI port.

Figure 117: The one-to-one mapping between MXK bridges and CPE Connections



Creating a one-to-one mapping between MXK bridge and CPE connections

The following example creates a one-to-one mapping which has one MXK bridge to mapping with one CPE connection.

- 1 Create a MXK bridge, and a CPE connection:

```
zSH> bridge add 1-3-1-5/gpononu gem 505 gtp 1 tm 1 downlink vlan 100 tagged eth 1 uni-vlan 100
Adding bridge on 1-3-1-5/gpononu
Created bridge-interface-record 1-3-1-505-gponport-100/bridge
CPE Connection 1-3-1-505/gponport/1/1/100/0 has been created
```

The first part of this command “**bridge add 1-3-1-5/gpononu gem 505 gtp 1 tm 1 downlink vlan 100 tagged**” creates a new MXK bridge. The second part of this command “**eth 1 uni-vlan 100**” creates a CPE connection with this MXK bridge. This example is passing single-tagged frames on VLAN 100 as is from Ethernet UNI port 1. For VLAN ID translation on ONU, refer to [VLAN translation on ONU, page 885](#).

GTP (GPON Traffic Profile) is a mandatory field in the **bridge add** command when creating a MXK bridge. It contains the bandwidth allocation information for the T-cont. For detail, refer to [Bandwidth Allocation for Upstream Traffic from the ONU to the MXK, page 925](#).

TM (Traffic Management Profile) is an optional field in the **bridge add** command when creating a MXK bridge. It contains the traffic shaping information for the GEM port. for detail, refer to [CPE Traffic Management Profiles, page 742](#).

- 2 Show MXK bridge and CPE connection.

```
zSH> bridge show
Orig
Type VLAN/SLAN VLAN/SLAN Physical Bridge St Table
Data-----
-----
dwn Tagged 100 1/3/1/5/gpononu 1-3-1-505-gponport-100/bridge DWN
```



```
upl          Tagged 100 1/a/5/0/eth    ethernet5-100/bridge    UP    S VLAN 100
default
2 Bridge Interfaces displayed
```

```
zSH> bridge show onu
```

ONU	GEM Port	ONU UNI	DSCP to COS	ONU UNI VLAN/SLAN	OLT VLAN/SLAN	OLT MVR	Service	OLT Bridge	ST
3/1/5	505	eth 1		100 Tagged	100		data	1-3-1-505-gponport-100/bridge	DWN

```
1 Bridge Interfaces displayed
1 GPON ONU Connections displayed
```

[Table 65](#) provides a summary on how to add bridges for supported services.

As shown in the table, for VoIP or PWE services, instead of **keyword + UNI port index**, the **bridge add** command specifies **keyword** only, which indicates the CPE connection is created on all the service-related UNI ports on this ONU.

Table 65: Dynamic bridging for services

Services	Bridges	Service Keywords	Command Examples
Data	One bridge add command per CPE connection	N/A Note: when no service keyword is specified, it implies data service.	bridge add 1-3-1-5/gpononu gem 301 gtp 1 downlink vlan 100 tagged eth 1 (It creates Data service on ethernet port 1 on the ONU)
Video	One bridge add command per CPE connection	video	bridge add 1-3-1-5/gpononu gem 401 gtp 1 video 0/4 downlink vlan 999 tagged eth 2 (It creates Video service on ethernet port 2 on the ONU)
VoIP	One bridge add command per ONU	sip, sipplar, or h248	bridge add 1-3-1-5/gpononu gem 702 gtp 1 downlink vlan 300 tagged sip (It creates a data path for SIP VoIP service on all POTS ports on the ONU) bridge add 1-3-1-5/gpononu gem 702 gtp 1 downlink vlan 300 tagged sipplar (It creates a data path for SIP PLAR VoIP service on all POTS ports on the ONU) bridge add 1-3-1-5/gpononu gem 702 gtp 1 downlink vlan 300 tagged h248 (It creates a data path for H.248 VoIP service on all POTS ports on the ONU)

Table 65: Dynamic bridging for services

Services	Bridges	Service Keywords	Command Examples
PWE	One bridge add command per ONU	pwe	bridge add 1-3-1-5/gpononu gem 602 gtp 1 downlink vlan 500 tagged pwe (It creates PWE service on all T1/E1 ports on the ONU)

All the command examples shown in [Table 65](#) are adding a VLAN to untagged traffics from ONU UNI ports. For VLAN ID translation, refer to [VLAN translation on ONU](#), page 885.

Deleting MXK bridge and associated CPE connections

The following example deletes a one-to-one mapping which has one MXK bridge mapping to one CPE connection.

To remove the MXK bridge and associated CPE connection at the same time:

```
zSH> bridge delete 1-3-1-505-gponport-100/bridge
CPE Connection 1-3-1-505/gponport/1/1/100/0 has been deleted
1-3-1-505-gponport-100/bridge delete complete
```

CPE Traffic Management Profiles

Users can use the CPE traffic management profile to configure the bandwidth shaping on GEM ports and Ethernet UNI ports (i.e. the Ethernet subscriber facing ports on an ONU). It is optional.

Note that the support of traffic management is ONU model dependent. Not all the ONU models can support traffic management.

To create a CPE traffic management profile, use this command:

```
cpe traffic add <profile-name>
[ us-sir < value > ]
[ us-pir < value > ]
[ ds-sir < value > ]
[ ds-pir < value > ]
[ us-priority < value > ]
[ us-weight < value > ]
[ ds-priority < value > ]
[ ds-weight < value > ]
```

Create a profile, the <profile-name> must be unique and the profile index will be automatically generated

Table 66: cpe traffic add Command Option Explanations

Command Option	Description
<i>profile-name</i>	Specify a unique name for the CPE traffic management profile. A profile index will be automatically generated after creation of this profile.
us-sir <i>value</i>	Upstream sustained information rate, in kilobits per second. Value range is 0 to 1310720.
us-pir <i>value</i>	Upstream peak information rate, in kilobits per second. Value range is 0 to 1310720.
ds-sir <i>value</i>	Downstream sustained information rate, in kilobits per second. Value range is 0 to 1310720. Only for Ethernet UNI ports.
ds-pir <i>value</i>	Downstream peak information rate, in kilobits per second. Value range is 0 to 1310720. Only for Ethernet UNI ports.
us-priority <i>value</i>	Upstream priority, for the strict priority scheduling policy. Value range is 0 to 7 where 0 is the highest priority.
us-weight <i>value</i>	Upstream weight, for the weighted round robin scheduling policy. Value range is 0 to 255 where 0 is the lowest weight.
ds-priority <i>value</i>	Downstream priority, for the strict priority scheduling policy. Value range is 0 to 7 where 0 is the highest priority.
ds-weight <i>value</i>	Downstream weight, for the weighted round robin scheduling policy. Value range is 0 to 255 where 0 is the lowest weight.

Configuring traffic shaping on GEM ports with CPE traffic management profiles

Users can configure the rate control on upstream, priority control on both upstream and downstream direction, and weight control on both direction of the GEM port.

Rate control and Priority (weight) control cannot be used on the same ONU. Users can either use the us-sir and us-pir fields in the CPE traffic management profile to define the upstream rate limiting on GEM ports, or use the us-priority, us-weight, ds-priority, ds-weight fields to define the priority and weight levels of upstream and downstream traffic on GEM ports.



Note: If rate control is applied to one GEM port, it will be enabled for the whole ONU, the other GEM ports in the same ONU must use rate control too. Same for Priority (Weight) control.



Note: Rate control on the downstream direction (i.e. ds-sir and ds-pir field in the CPE traffic management profile) only apply to Ethernet UNI ports. They do not apply to GEM ports.

The following example shows how to create the CPE traffic management profile, and associate it to a GEM port.

1 Create CPE traffic management profiles.

The following examples create two CPE traffic management profiles, one contains rate control parameters, the other one contains priority and weight control parameters.

Profile index is automatically created after the **cpe traffic add** command.

```
zSH> cpe traffic add 2MRateControl us-sir 2048 us-pir 2048
Profile "2MRateControl" has been created with index 1
```

```
zSH> cpe traffic add PriorityControl us-priority 2 us-weight 10 ds-priority 3
ds-weight 10
Profile "PriorityControl" has been created with index 2
```

```
zSH> cpe traffic show all
```

Index	Profile Name	---upstream---		--downstream--		---upstream---		--downstream--	
		SIR	PIR	SIR	PIR	priority	weight	priority	weight
1	2MRateControl	2048	2048	0	0	0	0	0	0
2	PriorityControl	0	0	0	0	2	10	3	10

2 entries found.

2 Associate tm 2MRateControl to GEM port 505 on ONU 3/1/5. The rate control parameters will apply to the GEM port.

The CPE traffic management profile can be referred to by either profile-name or profile-index.

```
zSH> bridge add 1-3-1-5/gpononu gem 505 gtp 1 tm 2MRateControl downlink vlan 100
tagged eth 1
Adding bridge on 1-3-1-5/gpononu
Created bridge-interface-record 1-3-1-505-gponport-100/bridge
CPE Connection 1-3-1-505/gponport/1/1/0/0 has been created
```

3 If users want to modify the CPE traffic management profile index of a GEM port, remove the GEM port with the **bridge delete** command first and then add it back with your desired CPE traffic management profile index.

```
zSH> bridge delete 1-3-1-505-gponport-100/bridge all
CPE Connection 1-3-1-505-gponport-100/bridge/1/1/100/0 has been deleted
1-3-1-501-gponport-100/bridge delete complete
```

This example associate traffic management profile PriorityControl to GEM port 505 on ONU 3/1/5. The priority control and weight control parameters will apply to the GEM port.

```
zSH> bridge add 1-3-1-5/gpononu gem 505 gtp 1 tm PriorityControl downlink vlan 100
tagged eth 1
Adding bridge on 1-3-1-5/gpononu
Created bridge-interface-record 1-3-1-505-gponport-100/bridge
```

Configuring rate control on Ethernet UNI ports with CPE traffic management profiles

Users can configure rate control on upstream and downstream directions of the Ethernet UNI port by using the us-sir, us-pir, ds-sir, and ds-pir fields in the CPE traffic management profile.



Note: Rate control on the downstream direction (i.e. ds-sir and ds-pir field in the CPE traffic management profile) only apply to Ethernet UNI ports. They do not apply to GEM ports.

The following example shows how to associate a CPE traffic management profile with a Ethernet UNI port.

- 1 Create a CPE traffic management profile TMethUNI, it contains the us-sir, us-pir, ds-sir, and ds-pir fields. Index 3 is assigned to it.

```
zSH> cpe traffic add TMethUNI us-sir 2048 us-pir 2048 ds-sir 3000 ds-pir 3000
Profile "TMethUNI" has been created with index 3
```

```
zSH> cpe traffic show all
```

Index	Profile Name	---upstream---		--downstream--		---upstream---		--downstream--	
		SIR	PIR	SIR	PIR	priority	weight	priority	weight
1	2MRateControl	2048	2048	0	0	0	0	0	0
2	PriorityControl	0	0	0	0	2	10	3	10
3	TMethUNI	2048	2048	3000	3000	0	0	0	0

3 entries found.

- 2 Associate CPE traffic management profile TMethUNI to Ethernet UNI port 1 on ONU 3/1/5. The rate control parameters defined in the us-sir, us-pir, ds-sir, and ds-pir fields will apply to the Ethernet UNI port.

Note that the CPE traffic management profile can be referred to by either profile-name or profile-index.

```
zSH> cpe eth add 3/1/5/1 admin-state up traffic-mngt-profile TMethUNI
```

- 3 If users want to modify the CPE traffic management profile index of an Ethernet UNI port, use the **cpe eth modify** command.

This example change the CPE traffic management profile index to 2MRateControl. This profile contains rate controls on the upstream only.

```
zSH> cpe eth modify 3/1/5/1 traffic-mngt-profile 2MRateControl
```

- 4 View the associated CPE traffic profile on an Ethernet port, use the **cpe eth show** command.

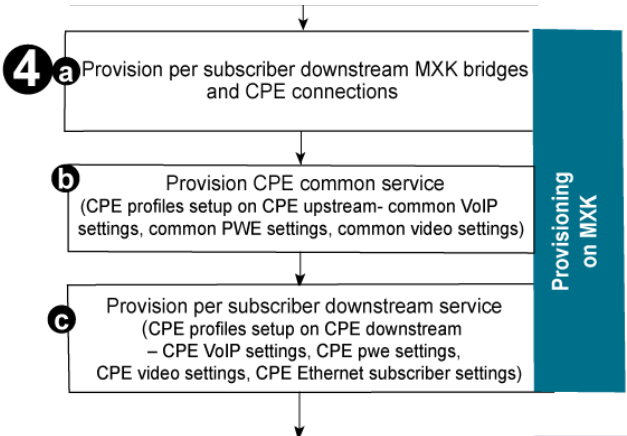
```
zSH> cpe eth show 3/1/5
```

CPE	Port Number	Admin	Rate	Duplex	Video Profile	Traf Mngt Profile	Alm	St	Sev
3/1/5	1	up	auto	auto	0	1	Dis	Mj	

1 services displayed

CPE Profiles

Figure 118: CPE profiles in Dynamic OMCII



CPE profiles define the different services that are provisioned on CPEs. As shown in the flowchart, there are two kinds of CPE profiles:

- CPE shared profiles (used in Step 4b)
CPE shared profiles contain the common service information which is used by multiple ONU UNI ports.



Note: The CPE shared profile can only be deleted if it is not associated with any other CPE profiles.

- CPE subscriber profiles (used in Step 4c)
CPE subscriber profiles contain the information for an individual ONU UNI port. When creating a CPE subscriber profile on an ONU UNI port, based on different services, users can associate related CPE shared profiles with it.
For example, when creating a CPE IP subscriber profile for VoIP service, a CPE IP common profile is required to be associated with it, if there is no CPE IP common profile index specified, the default CPE IP common profile will be used.

All the CPE profiles are listed in [Table 67](#).

As shown in the table, for Data or Video service, the creation of the CPE shared profiles and CPE subscriber profiles are not necessary, unless users want to change the default settings of those profiles. Creation of MXK bridges and CPE connections with the **bridge add** command are sufficient for creating a Data or Video service.

Table 67: CPE profile summary

Services	CPE Shared Profiles	CPE Subscriber Profiles
Data	N/A	CPE Eth profile (Optional)
Video	CPE video profile (Optional) CPE video access profile (Optional)	CPE Eth profile (Optional)
VoIP	CPE VoIP server profile CPE VoIP feature profile (Default profile provided. Only for SIP or SIP PLAR) CPE VoIP media profile (Default profile provided) CPE VoIP SIP dialplan profile (Optional. Only for SIP)	CPE VoIP profile (Requires CPE VoIP server profile, CPE VoIP feature profile, and CPE VoIP media profile)
	CPE IP common profile (Default profile provided)	CPE IP profile (Requires CPE IP common profile)
PWE	CPE PWE common profile (Default profile provided)	CPE PWE profile (Requires CPE PWE common profile)
	CPE IP common profile (Default profile provided)	CPE IP profile (Requires CPE IP common profile)
RF	N/A	CPE RF profile

The step-by-step configuration procedure for each service are provided in [OMCI GPON zNID Installation with Dynamic OMCI for Triple Services on page 748](#).

CPE UNI Ports Default States

If a CPE is managed by MXK, a CPE UNI is disabled until service is created on the UNI. However, service can also be created by the other management tools through CPE manager or through pre-configuration. In that case, the UNI needs to be enabled explicitly.

This example enabled Ethernet port 1 on CPE 2/3/4:

```
zSH>CPE>ETH> add 2/3/4/1 admin-state up
```

OMCI GPON zNID Installation with Dynamic OMCI for Triple Services

In this section, users will provision Data service, Video service, VoIP service, and PWE service on the same ONU, just the MXK bridge interface, GEM port setup, GPON traffic profile, VLAN, UNI ports are different. And provision RF service on another ONU. For ease of discussion each of the applications is described separately in this section.

Generally these are the steps to follow to configure the MXK to be able to manage OMCI GPON zNIDs with Dynamic OMCI:

- [Create High Speed Internet on Dynamic OMCI with Uplink and Downlink, page 748](#)
- [Create Uplink and Downlink Bridges on Dynamic OMCI for Video, page 755](#)
- [Create VoIP on Dynamic OMCI on Uplink and Downlink Bridges, page 763](#)
- [Create PWE on Dynamic OMCI on Uplink and Downlink Bridges, page 779](#)
- [Create RF on Dynamic OMCI, page 787](#)

Create High Speed Internet on Dynamic OMCI with Uplink and Downlink

- [Creating a GPON traffic profile, page 748](#)
- [Specifying internal ME profile for the ONU, page 749](#)
Only need to specify the internal ME profile once for each ONU.
- [Creating uplink/downlink MXK bridges, and CPE connections, page 750](#)
- [Creating CPE Ethernet subscriber profile \(optional\), page 751](#)
- [Activating the ONT, page 753](#)
Only need to activate the ONU once.
- [Testing the data bridge, page 755](#)

In this example, for data service users will create uplink/downlink bridges with VLAN 1001.

To create data service on the Ethernet UNI ports, use the following steps:

Creating a GPON traffic profile

The GPON traffic profile is a template for defining how traffic will be handled on the bridge with which the GTP is associated. One GTP may be associated with many different bridges. The GTP in this procedure will create a high bandwidth configuration.

Refer to [Configure GPON traffic profile, page 925](#) to get detail configuration and parameter description.

The following is recommended for high speed data configurations.

```
zSH> new gpon-traffic-profile 1
gpon-traffic-profile 1
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}: 2048
traffic-class: -----> {ubr}:      ubr is the default value
compensated: -----> {false}:
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: -----> {0}:
dba-fixed-us-cbr-bw: -----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Specifying internal ME profile for the ONU

Based on the ONU model, assign the matching internal ME profile to this ONU. In this way it indicates this ONU is provisioned by dynamic OMCI.

1 List the currently valid internal ME profiles.

```
zSH> gpononu profile show internal-me
zhone-1e          1 ETH
zhone-2301        1 GE
zhone-2311        1 GE
zhone-2402        2 GE
zhone-2403        2 GE + 1 RFV)
zhone-2424        4 GE + 2 POTS
zhone-2425        4 GE + 2 POTS + 1 RFV
zhone-2426        4 GE + 2 POTS + 1 WiFi(wlan1-wlan4) + 1 USB
zhone-2427        4 GE + 2 POTS + 1 WiFi(wlan1-wlan4) + 1 RFV + 1 USB
zhone-2501        1 GE
zhone-2504        4 GE
zhone-2510        4 FE + 2 POTS
zhone-2510a       4 FE + 2 POTS
zhone-2511        4 FE + 2 POTS + 1 RFV
zhone-2516        4 GE + 2 POTS + 2 WLAN
zhone-2517        4 GE + 2 POTS + 2 WLAN + 1 RFV
zhone-2520        4 FE + 4 POTS
zhone-2543        4 GE + 2 POTS + 1 RFV
zhone-2608t       8 GE PoE
zhone-2624a       4 GE + 2 POTS
zhone-2624p       4 GE PoE + 2 POTS
zhone-2628a       8 GE + 2 POTS
zhone-2628p       4 GE PoE(eth1-eth4) + 4 GE(eth5-eth8) + 2 POTS
zhone-2628t       8 GE PoE + 2 POTS
zhone-2644a       4 GE + 4 POTS
```

zhone-2644p	4 GE PoE + 4 POTS
zhone-2648a	8 GE + 4 POTS
zhone-2648p	4 GE PoE(eth1-eth4) + 4GE(eth5-eth8) + 4 POTS
zhone-2648t	8 GE PoE + 4 POTS
zhone-2802p	2 GE PoE
zhone-2804p	4 GE PoE
zhone-4220	2 GE + 2 POTS + 1 USB
zhone-4221	2 GE + 2 POTS + 1 RFV + 1 USB
zhone-4222	2 GE + 1 HCNA(eth3) + 2 POTS + 1 USB
zhone-4223	2 GE + 1 HCNA(eth3) + 2 POTS + 1 RFV + 1 USB
zhone-4224	2 GE + 1 HCNA(eth3) + 1 HPNA(eth4) + 2 POTS + 1 USB
zhone-4225	2 GE + 1 HCNA(eth3) + 1 HPNA(eth4) + 2 POTS + 1 RFV + 1 USB
zhone-4226	6 GE + 2 POTS + 1 USB
zhone-4240	2 GE + 4 POTS + 1 USB
zhone-4241	2 GE + 4 POTS + 1 RFV + 1 USB
zhone-4242	2 GE + 1 HCNA(eth3) + 4 POTS + 1 USB
zhone-4243	2 GE + 1 HCNA(eth3) + 4 POTS + 1 RFV + 1 USB
zhone-4244	2 GE + 1 HCNA(eth3) + 1 HPNA(eth4) + 4 POTS + 1 USB
zhone-4222a	2 GE + 2 POTS + 1 USB
zhone-4224a	4 GE + 2 POTS + 1 USB
zhone-4222h	2 GE + 1 HCNA(eth3) + 2 POTS + 1 USB
zhone-4224h	4 GE + 1 HCNA(eth5) + 2 POTS + 1 USB
zhone-5114	4 GE + 2 POTS + 4 T1/E1
zhone-5120	4 GE + 2 POTS + 8 T1/E1
zhone-7310	1 FE + 8 POTS + 2 T1/E1
zhone-8224	24 FE
zhone-8324	24 FE + 24 POTS
zhone-8424	24 FE
zhone-8524	24 FE + 24 POTS
zhone-9108	9 GE PoE + SFP(eth10) + 8 POTS + 1 USB
zhone-9208	9 GE PoE + SFP(eth10) + 8 POTS + 8 RF(MOCA) + 1 USB
zhone-9308	9 GE PoE + SFP(eth10) + 8 POTS + 8 RF(MOCA+RFV) + 1 USB
zhone-9440	5 GE PoE + SFP(eth10) + 4 T1/E1 + 1 USB
zhone-9444	5 GE PoE + SFP(eth10) + 4 POTS + 4 T1/E1 + 1 USB
zhone-9480	9 GE PoE + SFP(eth10) + 8 T1/E1 + 1 USB
zhone-9488	9 GE PoE + SFP(eth10) + 8 POTS + 8 T1/E1 + 1 USB
zhone-cig	24 FE + 24 POTS
zhone-default	24 FE + 24 POTS + 24 T1/E1 + 24 RFV

2 Specify an internal ME profile for an ONU.

After specifying internal ME profile zhone-5114 for ONU 1/3/1, dynamic OMCI supports are associated with the ONT.

```
zSH> onu set 1/3/1 meprof zhone-5114
```

Creating uplink/downlink MXK bridges, and CPE connections

This section will create an uplink and downlink bridges for VLAN 1001 on MXK, and create CPE connections for the same VLAN on two ONU Ethernet UNI ports:

1 Create an uplink bridge interface on the MXK

```
zSH> bridge add 1-a-8-0/eth uplink vlan 1001
Adding bridge on 1-a-8-0/eth
Created bridge-interface-record ethernet8-1001/bridge
bridge-path added successfully
```

2 Create downlink bridges on the MXK and CPE connections on ONU Ethernet UNI ports.

This example shows ONU Ethernet UNI port 1 map to the GEM port 610 and VLAN 1001.

Create an MXK bridge on GEM port 610, and a CPE connection on ONU Ethernet UNI port 1. Associate GTP 1 with GEM port 610. Both the CPE connection and the GEM port are in the same VLAN 1001.

```
zSH> bridge add 1-1-3-1/gpononu gem 610 gtp 1 downlink vlan 1001 tagged eth 1
uni-vlan 1001
Adding bridge on 1-1-3-1/gpononu
Created bridge-interface-record 1-1-3-610-gponport-1001/bridge
CPE Connection 1-1-3-610/gponport/1/1/1001/0 has been created
```

3 View the MXK bridges.

```
zSH> bridge show
Orig
Type VLAN/SLAN VLAN/
SLAN Physical Bridge St Table Data
-----
dwn Tagged 1001 1/1/3-1/gpononu 1-1-3-610-gponport-1001/bridge UP
upl Tagged 1001 1/a/8-0/eth ethernet8-1001/bridge UP S
VLAN 1001 default
2 Bridge Interfaces displayed
```

4 View the CPE connections.

```
zSH> bridge show onu
GEM ONU DSCP ONU UNI OLT OLT
ONU Port UNI to COS VLAN/SLAN VLAN/SLAN MVR Service OLT Bridge ST
-----
1/3/1 610 eth 1 1001/ Tagged 1001 data 1-1-3-610-gponport-1001/bridge DWN
1 Bridge Interfaces displayed
1 GPON ONU Connections displayed
```

Creating CPE Ethernet subscriber profile (optional)

By default, the admin-state of the CPE Ethernet UNI port is up after creation of CPE connection on that CPE Ethernet UNI port. Because of that, the Data and Video traffic can run on this Ethernet UNI port without further configuration.

If users want to change the default Ethernet physical configurations, users can use the **cpe eth add** command. With this command, the CPE Ethernet subscriber profile is created *manually*. The Ethernet UNI port ID specified in this command must match the one assigned in the **bridge add** command when creating downlink bridge and CPE connection.

Note that a CPE ethernet subscriber profile will also be created *automatically* if users set the ONU UNI ethernet port's admin-state with the **gpononu port** command, or modify the port speed with the **gpononu auto-detect** command. For the details on those command, refer to [Administration of subscriber facing ports on page 914](#), and [Configurable speed of subscriber facing ports on page 915](#).



Note: It is recommended to keep the default settings of the CPE ethernet subscriber profile. It might have an impact on the MXK system if there are too many CPE ethernet subscriber profiles.

After a CPE ethernet subscriber profile is created, if users want to change the settings in that profile, use the **cpe eth modify** command, which has the same command syntax as the **cpe eth add** command. Only change it when it is necessary.

Command:

```
cpe eth add <interface>/<port number>
    [ admin-state < up | down > ]
    [ rate < auto | 10 | 100 | 1000 > ]
    [ duplex < auto | full | half > ]
    [ video-profile < index | profile-name > ]
    [ traffic-mngt-profile < index |
profile-name > ]
    [ line-status-alarm < enabled | disabled>
]
    [ alarm-severity < critical | major |
minor | warning >
```

Create a ETH service. <interface> and <port number> must be provided.

[Table 96](#) provides the description for command options in the **cpe eth add** command.

Table 68: cpe eth add Command Option Explanations

Command Option	Description
<i>interface/port number</i>	ONU port ID and Ethernet UNI port ID of the physical interfaces.
admin-state <i>value</i>	Activates or deactivates the functions performed by the Ethernet port for this subscriber. Possible values are up, down. Default value is up.
rate <i>value</i>	Sets the Ethernet port rate. Possible values are auto (default), 10, 100, 1000.
duplex <i>value</i>	Sets the Ethernet port duplex. Possible values are auto (default), full, half.

Table 68: cpe eth add Command Option Explanations

Command Option	Description
video-profile <i>index profile-name</i>	Associated CPE video profile. Default is 0.  Note: The video-profile field is only for video service.
traffic-mgmt-profile <i>index profile-name</i>	Associated CPE traffic management profile. Default is 0.
line-status-alarm <i>value</i>	Enables or disables line status alarms on this port. Possible values are enabled or disabled. Default is disabled.
alarm-severity <i>value</i>	Sets the severity of line status alarms on this port. The severity level takes effect only after line-status-alarm has been enabled. Possible values are critical, major, minor, warning. Default value is major.

To create a CPE Eth subscriber profile with the **cpe eth add** command:

- 1 This example changed the Ethernet rate and duplex mode of the Ethernet UNI port 1 on the ONU 1/3/1. Note that this example enters CPE command shell: zSH> CPE> ETH>.

```
zSH> CPE> ETH> add 1/3/1/1 rate 100 duplex full
Service has been created
```

- 2 Show the settings of the CPE Eth Profile for the Ethernet UNI port 1 on the ONU 1/3/1.

```
zSH> CPE> ETH> show 1/3/1/1
```

CPE	Port Number	Admin	Rate	Duplex	Video Profile	Traf Mngt Profile	Alm St	Sev	Power Shed	Power Range
1/3/1	1	up	100	full	0	0	Dis	Mj	Dis	Medium

1 services displayed

- 3 Show all the services created for the ONU.

```
zSH> CPE> ETH> exit
zSH> CPE> show 1/3/1
```

CPE 1/3/1

Service: DATA

GEM	UNI	UNI	VLAN/SLAN	OLT	VLAN/SLAN	G-VLAN	Admin	Oper	Rg-Mode
610	eth 1	1001/----	Tagged	8,1001	0	up			

Activating the ONT

Activate the ONT to add it to the system. If users are adding multiple services, users would range the ONT after all the services have been added. an

activated ONT is an ONT had assigned a serial number on, and the ONT port admin status is up.

In Dynamic OMCI, after changing the service configuration on an activated ONU, the services configuration will be updated automatically.

- 1 To activate an ONT first run the **gpononu show** command to display the ONTs currently on the OLT, and discover the available serial numbers.

The **gpononu show** command has options to select by slot and OLT. If users run the command without defining the slot/OLT the command will check for ONTs on every port of every card and depending on the number of cards, may take several minutes to complete.

```
zSH> gpononu show 1/3
```

```
Free ONUs for slot 1 olt 3:
```

```

 1    2    3    4    5    6    7    8    9   10   11   12
13   14   15   16   17   18   19   20   21   22   23   24
25   26   27   28   29   30   31   32   33   34   35   36
37   38   39   40   41   42   43   44   45   46   47   48
49   50   51   52   53   54   55   56   57   58   59   60
61   62   63   64
```

```
Discovered serial numbers for slot 1 olt 3:
```

```

sernoID  Vendor  Serial Number      Time Discovered
 1       ZNTS    93425008           JUL 10 01:11:18 2014
```

- 2 Run the **gpononu set SlotID/OltID/OnuID** command to associate a ONU port ID to a discovered ONT's serial number.

```
zSH> gpononu set 1/3/1 1
```

```

Onu 1 successfully enabled with serial number ZNTS
93425008
```

- 3 Run the **gpononu show** command to verify the ONT is enabled, and OMCI support is added into the ONT (the associated internal ME profile can be displayed).

```
zSH> gpononu show 1/3/1
```

```

                        Serial
ONU      Name          Enabled Model # Number      OMCI files and profiles
=== =====
 1       1-1-3-1      Yes      5114 ZNTS 93425008      ME   zhone-5114
```

```
Note: NULL Model String indicates not able to get model ID
```

- 4 Run the **gpononu status** command to verify the OMCI Config State is active.

```
zSH> gpononu status 1/3/1
```

```

Download  OLT      ONT      Distance  GPON
ID   Onu   OperStatus ConfigState  State  Rx Power  Rx Power (KM)  OnuStatus
==  =====
1   1-1-3-1   Up      Active      NoUpgrade  -19.2 dBm -20.0 dBm   18      Active
```

- 5 Run the **port show** command to verify the ONT port admin status is up.

```
zSH> port show 1-1-3-1/gpononu
```

```
Interface 1-1-3-1/gpononu
Administrative status: up
```

- 6 If users want to remove the serial number assignment from the ONT, use the **onu clear SlotID[/OltID[/OnuID]]** command. If users want to remove the OMCI profiles as well, use the keyword **omci** in the command.

In addition, in the **onu clear** command, the Slot ID, OLT ID, and ONU ID maybe replaced with brackets containing numbers in comma-separated series (e.g. [1,4]), in dash-separated ranges (e.g.[1,3-4]), and OLT ID and ONU ID in wildcard (i.e. not specifying OLT ID or ONU ID).

```
zSH> gpononu clear 1/3/1
Onu 1 (previously with serial number ZNTS 93425008 ) has been cleared
```

Testing the data bridge

Verify the user can get data on the PC:

- 1 Connect an ONT downlink ethernet port to a PC.

Make sure the ONT model matches the internal ME profile users assigned. This example connects a ZNID-GPON-5114 to the PC.

And also make sure the ONT downlink ethernet port number matches the one users assigned with the **bridge add** command for data service. In this example, users can connect either ETH 1 or ETH 2 to the PC.

- 2 Run the **bridge show** command to view the MAC address of the connected PC.

```
zSH> bridge show
      Orig
Type VLAN/SLAN VLAN/SLAN Physical      Bridge      St Table Data
-----
upl      Tagged 1001 1/a/8/0/eth ethernet8-1001/bridge      UP S VLAN 1001 default
dwn      Tagged 1001 1/1/3/1/gpononu 1-1-3-610-gponport-1001/bridge
UP D 00:00:86:43:3c:e4 MAC of PC
```

- 3 Open a command prompt on the PC and enter **ipconfig** to verify that users can get an IP address from DHCP server for the PC.
- 4 Open an internet browser on the PC, users should be able to access the internet now.

Create Uplink and Downlink Bridges on Dynamic OMCI for Video

Video bridging is very similar to data bridging, it uses downlink and uplink bridges as well, but the GTP, GEM ports and VLANs are different. During configuring data service, the internal ME profile name is already specified to the ONU, so if the video service is going to be configured on the same ONU, there is no need to specify the internal ME profile again.

To create video service on the Ethernet UNI ports on the same ONU use the following steps:

- [Creating GPON traffic profile, page 756](#)
- [Creating uplink and downlink bridges, and CPE connections, page 756](#)

Creation of Video bridge in this step is sufficient for creating a video service. If users want to configure additional common video service attributes and subscriber attributes, use the following optional steps.

- [Creating CPE video access control \(optional\), page 758](#)
- [Creating CPE video profile and associate it with a CPE video access control list \(optional\), page 760](#)
- [Creating CPE Ethernet subscriber profile and associate it with a CPE video profile \(optional\), page 762](#)
- [Testing the IPTV bridge, page 762](#)

Creating GPON traffic profile

Add the GPON traffic profile.

The following GPON traffic profile is recommended for video:

```
zSH> new gpon-traffic-profile 2
gpon-traffic-profile 2
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}: 512
traffic-class: -----> {ubr}:
compensated: -----> {false}:
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: -----> {0}:
dba-fixed-us-cbr-bw: -----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.qq

g
```

Creating uplink and downlink bridges, and CPE connections

This section will create an uplink and downlink bridge for VLAN 999:

1 Create an uplink bridge interface

a Create the uplink bridge interface

The following example creates a video uplink bridge interface, and enables IGMP proxy with IGMP snooping.

```
zSH> bridge add 1-a-4-0/eth uplink vlan 999 tagged
igmpproxy
```



```
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record 1-a-4-0-eth-999/
bridge
Bridge-path added successfully
```

- b** Modify the bridge-path for the uplink with 30 seconds IGMP query interval. Note how the igmptimer is added to the bridge-path.

```
zSH> bridge-path modify 1-a-4-0-eth-999/bridge vlan
999 default igmptimer 30
```

- 2** Create downlink bridge interfaces and creates CPE connections on ONU Ethernet UNI ports.

Create downlink bridges on a GEM port and ONU with VLAN ID and GTP.

Users can also specify **video m/n**. *m* indicates the multicast control list on the *MXK bridge*, *n* indicates the maximum video streams on the MXK bridge. The maximum video streams is a mandatory field when users creating a downlink bridge for video service, otherwise the bridge will fail to pass all video mulitcast traffic. By specifying **video 0/4** in this example users can enable subscriptions up to four video streams on the MXK bridge interface without control list checking.

If users want to control multicast control list checking on the *CPE connection*, use the **CPE video access add** command to create CPE video access control profiles.

This example adds VLAN 999 to video traffic.

```
zSH> bridge add 1-1-3-1/gpononu gem 650 gtp 2 downlink cos 6 vlan 999 tagged video
0/4 eth 3
```

```
Adding bridge on 1-1-3-1/gpononu
Created bridge-interface-record 1-1-3-650-gponport-999/bridge
CPE Connection 1-1-3-650/gponport/12/3/0/0 has been created
```

- 3** View the MXK bridges.

```
zSH> bridge show
```

Orig	Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
dwn	Tagged	999	1/1/3/1/gpononu	1-1-3-650-gponport-999/bridge	UP			
dwn	Tagged	1001	1/1/3/1/gpononu	1-1-3-610-gponport-1001/bridge	UP			
upl	Tagged	999	1/a/4/0/eth	ethernet4-999/bridge	UP	S	VLAN 999	default
upl	Tagged	1001	1/a/8/0/eth	ethernet5-1001/bridge	UP	S	VLAN 1001	default

4 Bridge Interfaces displayed

- 4** View the CPE connections.

```
zSH> bridge show onu
```

ONU	GEM	ONU	DSCP	ONU UNI	OLT	OLT	Service	OLT Bridge	ST
Port	Port	UNI	to COS	VLAN/SLAN	VLAN/SLAN	MVR			
1/3/1	610	eth 1		1001/	Tagged 1001		data	1-1-3-610-gponport-1001/bridge	UP
1/3/1	650	eth 3			Tagged 999		iptv	1-1-3-650-gponport-999/bridge	UP

2 Bridge Interfaces displayed
2 GPON ONU Connections displayed

5 View the services on the CPE.

```
zSH> CPE> show 1/3/1

CPE 1/3/1
Service: DATA
GEM   UNI   UNI VLAN/SLAN OLT VLAN/SLAN Admin Oper   Rg-Mode
-----
 610  eth 1   1001/---- Tagged 1001   up   down

Service: IPTV
GEM   UNI   UNI VLAN/SLAN OLT VLAN/SLAN Admin Oper   Video Prof
-----
 650  eth 3           Tagged 6, 999   up   down           1 6 is the Video CoS
value
```

Creating CPE video access control (optional)

CPE video access control profile creates an access control list, which defines which multicast addresses the remote-end video can access. That access control list can be specified later in the CPE video profile. If there is no CPE video access control profiles specified in the CPE video profile, there will be no control on the multicast addresses.



Note: The CPE video access control profile can not be deleted if this profile is the only entry in an access control list that is being associated with a CPE video profile.

To create a new CPE video access profile, use this command:

Command:

```
cpe video access add <list-name>      [ type <normal |
always-on | periodic> ]
    [ src-ip <IPAddress>]
    [ dest-ip-start <IPAddress>]
    [ dst-ip-end <IPAddress> ]
    [ imputed-group-bw <value in bytes/sec>]
```

Table 69 provides the description for command options in the **cpe video access add** command.

Table 69: cpe video access add Command Option Explanations

Command Option	Description
<i>list-name</i>	CPE video access control profile name.
type	Defines the video stream type. Possible values are normal (default), always-on, periodic.

Table 69: cpe video access add Command Option Explanations

Command Option	Description
src-ip <i>IPAddress</i>	Source IP address. The default value is 0.0.0.0, that indicates that source IP address is to be ignored.
dst-ip-start <i>IPAddress</i>	Destination IP address of the start of the multicast range.
dst-ip-end <i>IPAddress</i>	Destination IP address of the end of the multicast range.
imputed-group-bw <i>value</i>	Imputed group bandwidth. In the unit of bytes/second. The imputed group bandwidth is used to decide whether or not to honor a join request in the presence of a max multicast bandwidth limit. The default value 0 effectively allows this table entry to avoid maximum bandwidth limitations.

This example creates two CPE video access control profiles, each profile is an entry of a CPE video access control list:

- 1 Create a CPE video access control profile.

```
zSH> CPE> VIDEO> ACCESS> add basic-plan dst-ip-start
224.10.10.1 dst-ip-end 224.10.10.15 imputed-group-bw
4000
Profile has been created with index 1/1
```

The first CPE video access control profile in the system is created automatically with list-index 1/entry-index 1.

- 2 Create the second CPE video access control profile under the same list (1/2):

```
zSH> CPE> VIDEO> ACCESS> add basic-plan dst-ip-start
224.11.10.1 dst-ip-end 224.11.10.4 imputed-group-bw
4000
Profile has been created with index 1/2
```

- 3 View the two cpe-video-access-control profiles that under the same list. users can either specify list-name or list-index.

```
zSH> CPE> VIDEO> ACCESS> show 1
List/Entry Index      Profile Name          dstIpStart      dstIpEnd      imputedGroupBw
-----
1/1                  basic-plan 224.10.10.1 224.10.10.15    4000
1/2                  basic-plan 224.11.10.1 224.11.10.4     4000
2 entries found.
```

- 4 If users want to delete a cpe-video-access-control profile, use the **cpe video access delete** <list-name | list-index> / <entry-index> command.

```
zSH> CPE> VIDEO> ACCESS> delete 1/1
Profile has been deleted.
```

- 5 If users want to delete the last CPE video access control profile in an access control list that is being associated with a CPE video profile. Users have to remove the reference in the CPE video profile first, and then delete the CPE video access control profile.

This example assumes access control list 1 is being associated with CPE video profile 1, and CPE video access control profile 1/2 is the only entry in the access control list 1.

a Cannot delete CPE video access control profile 1/2.

```
zSH> CPE> VIDEO> ACCESS> delete 1/2
Error: Cannot delete cpe-video-access-control profile
Cannot delete last entry in a list that is being used by a cpe-video profile.
```

b Find associated CPE video profiles with access control list 1.

```
zSH> CPE> VIDEO> ACCESS> find 1
cpe-video 1/4095/0
1 profiles displayed.
```

c Show the detail information of the associated CPE video profile.

```
zSH> CPE> VIDEO> show 1
```

Index	Profile Name	Max-Simultaneous Groups	Max Mcast Bw	Bw Enforce	Access Control List
1	basic-plan	4	50000	true	1

1 entries found.

d Remove the reference of the access control list from the CPE video profile. The command option of the **cpe video modify** command is same as **cpe video add**, refer to the next section for the detail.

```
zSH> CPE> VIDEO> modify 1 access-control-list 0
Profile has been modified.
```

e Now users can delete CPE video access control profile 1/2.

```
zSH> CPE> VIDEO> ACCESS> delete 1/2
Profile has been deleted.
```

Creating CPE video profile and associate it with a CPE video access control list (optional)

This step is optional. Creation of Video bridge in step [Creating uplink and downlink bridges, and CPE connections on page 756](#) is sufficient for creating a video service. Users can use the cpe-video-access-control profile and cpe-video profile, if users want to add some common video service attributes.



Note: CPE video profile can only be deleted when it is not associated with any CPE ethernet subscriber profiles.

Command:

```
cpe video add <profile-name <string>>
[ max-simultaneous-groups < value > ]
[ max-mcast-bw < value > ]
[ bw-enforce < value > ]
```

[**access-control-list** < value >]

Table 70 provides the description for command options in the **cpe video add** command.

Table 70: cpe video add Command Option Explanations

Command Option	Description
<i>profile-name</i> <string>	Specifies a unique CPE video profile name. 36 characters string.
max-simultaneous-groups <i>value</i>	Specifies the maximum number of dynamic multicast groups that may be joined by at any one time. Default: 0 . Specifies that no administrative limit is to be imposed.
max-mcast-bw <i>value</i>	Specifies the maximum imputed dynamic bandwidth, in bytes per second, that may be delivered to the client port at any one time. Default: 0 . Specifies that no administrative limit is to be imposed
bw-enforce <i>value</i>	Values: true Specifies that such attempts be counted and denied. The imputed bandwidth value is taken from the dynamic access control list table, both for a new join request and for pre-existing groups. false Specifies that attempts to exceed the max multicast bandwidth be counted but honored. Default: false
access-control-list <i>value</i>	This attribute points to a access control group list. Default: 0 . It indicates no control list is used.

1 Create the cpe-video profile.

And users can also associate a video access control list to it.

```
zSH> CPE> VIDEO> add basic-plan max-simultaneous-groups 4 max-mcast-bw 50000
bw-enforce true access-control-list basic-plan
Profile "basic-plan" has been created with index 1
```

2 Show all the cpe-video profiles.

```
zSH> CPE> VIDEO> show all
```

Index	Profile Name	Max-Simultaneous Groups	Max Mcast Bw	Bw Enforce	Access Control List
1	basic-plan	4	50000	true	1

1 entries found.

3 If users want to delete a cpe video profile, use the **cpe video delete** <profile-index> | <profile-name> command.

```
zSH> CPE> VIDEO> delete 1
Profile has been deleted.
```

4 Users can use the find command to find the associated CPE Ethernet subscriber profile.

This example assumes CPE video profile 1 is being associated with a CPE ethernet subscriber profile on ONU 1/3/1:

```
zSH> CPE> VIDEO> find 1
cpe-eth-subscriber 1-1-3-1/gpononu
1 profiles displayed
```

Creating CPE Ethernet subscriber profile and associate it with a CPE video profile (optional)

If users want to configure subscriber video service attributes on the Ethernet UNI port, users can use the cpe-eth-subscriber profile. In the cpe-eth-subscriber profile users can change the default Ethernet physical configurations, such as loopback, rate, duplex, or assign a video profile to the Ethernet UNI port. If video profile field is not specified or specified to index 1, the default CPE video profile will be used.

For general information and value description of the CPE Ethernet subscriber profile, refer to [Creating CPE Ethernet subscriber profile \(optional\) on page 751](#).

To create a CPE ether subscriber profile for video service:

- 1 Create a CPE Eth subscriber profile on ONU 1/3/1 Ethernet UNI port 3, and associate it with cpe-video profile “basic-plan”.

Make sure the Ethernet UNI port matches the port ID assigned during the creation of the downlink bridge and CPE connection.

```
zSH> CPE> ETH> add 1/3/1/3 video-profile basic-plan
Service has been created
```

- 2 Show CPE Eth subscriber profile 1/3/1/3.

```
zSH> CPE> ETH> show 1/3/1/3
```

CPE	Port Number	Admin	Rate	Duplex	Video Profile	Traf Mngt Profile	Alm St	Sev	Power Shed	Power Range
1/3/1	3	up	auto	auto	1	0	Dis	Mj	Dis	Medium

1 services displayed

Testing the IPTV bridge

Since users are using a PC and software to emulate a set top box (STB), users can ping out to the video server.

- 1 Connect an ONT downlink ethernet port to a customer video equipment. This example connects to a PC that runs a STB emulation software.

Make sure the ethernet port number matches the one users assigned with the **bridge add** command for video service. In this example users can connect ETH UNI port 3 to the PC.
- 2 Open a command prompt on the PC and enter **ipconfig** to verify that users can get an IP address for the PC.

3 Ping the video server**a** Open a DOS window**b** Ping the upstream gateway (provided in your environment setup)**4 Open the STB emulation software and connect to the video server**

As long as users can ping that means there are a data path through the zNID and the MXK to the video server. Users should be able to connect to the video stream with the STB emulation software.

Create VoIP on Dynamic OMCI on Uplink and Downlink Bridges

For VoIP service Zhone recommends using uplink and downlink bridge configuration.

There are different types of VoIP signaling: SIP, SIP PLAR, H.248, or MGCP. One ONU can only use one VoIP signaling protocol.



Note: One ONU only can have one VoIP signaling. For example, if users configured POTS 1 for SIP, all the POTS ports in the same ONU must use SIP too.



Note: Make sure country code in the system profile is set properly for the voice signaling type.

To create voice service on the POTS ports on the same ONU, use the following steps:

- [Creating GPON traffic profile, page 764](#)
- [Creating the uplink and downlink bridge and CPE connection, page 764](#)
- [Creating a CPE IP common profile for VoIP \(Optional\), page 765](#)
- [Creating a CPE IP profile for the VoIP service and associate it with a CPE IP common profile, page 766](#)
- [Creating CPE VoIP server profiles, page 767](#)
- [Creating CPE SIP dial plans for a SIP VoIP server \(optional\), page 771](#)
This profile is only needed for SIP voice signaling.
- [Creating CPE VoIP features profile for SIP or SIP PLAR, page 772](#)
This profile is only needed for SIP or SIP PLAR voice signaling.
- [Creating CPE VoIP media profile, page 775](#)
- [Creating a CPE VoIP subscriber profile and associate it with a VoIP server, a VoIP features profile, and a media profile, page 776](#)
- [Testing the VoIP configuration, page 779](#)

Creating GPON traffic profile

Add the GPON traffic profile.

The following GPON traffic profile is recommended for up to four VoIP phones or four POTS ports.

```
zSH> new gpon-traffic-profile 3

gpon-traffic-profile 3
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}: 512
traffic-class: -----> {ubr}:
compensated: -----> {false}:
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: -----> {0}:
dba-fixed-us-cbr-bw: -----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved..
```

Creating the uplink and downlink bridge and CPE connection

This section will create a uplink bridge and a downlink bridge on the MXK and create CPE connection on the ONU for SIP service:

- 1 Create an uplink bridge on the uplink interface.

```
zSH> bridge add 1-a-4-0/eth uplink vlan 300 tagged
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-300/bridge
Bridge-path added successfully
```

- 2 Create a downlink bridge on the downlink interface and create a CPE connection for all POTS ports on the ONU.

This example creates a downlink MXK bridge on GEM port 710, and creates SIP service for all the POTS ports on ONU 1/3/1 and inserts single tagged VLAN 300 to the untagged SIP packets.

```
zSH> bridge add 1-1-3-1/gpononu gem 710 gtp 3 downlink vlan 300 tagged sip
Adding bridge on 1-1-3-1/gpononu
Created bridge-interface-record 1-1-3-710-gponport-300/bridge
CPE Connection 1-1-3-710/gponport/14/0/0/0 has been created
```

- 3 On MXK, run the **bridge show** command to view the MXK bridges.

```
zSH> bridge show
```

Orig Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
dwn	Tagged	300	1/1/3/1/gpononu	1-1-3-710-gponport-300/bridge	UP		


```

dwn          Tagged 999 1/1/3/1/gpononu 1-1-3-650-gponport-999/bridge UP
dwn          Tagged 1001 1/1/3/1/gpononu 1-1-3-610-gponport-1001/bridge UP
upl          Tagged 300 1/a/4/0/eth ethernet4-300/bridge UP
upl          Tagged 999 1/a/4/0/eth ethernet4-999/bridge UP S VLAN 999 default
upl          Tagged 1001 1/a/8/0/eth ethernet8/bridge UP S VLAN 1001 default
6 Bridge Interfaces displayed

```

4 View the CPE connections.

```
zSH> bridge show onu
```

ONU	GEM Port	ONU UNI	DSCP to COS	ONU UNI VLAN/SLAN	OLT VLAN/SLAN	OLT MVR	Service	OLT Bridge	ST
1/3/1	610	eth 1		1001/----	Tagged 1001		data	1-1-3-610-gponport-1001/bridge	UP
1/3/1	650	eth 3			Tagged 999		iptv	1-1-3-650-gponport-999/bridge	UP
1/3/1	710	pots			Tagged 300		sip	1-1-3-710-gponport-300/bridge	UP

```

3 Bridge Interfaces displayed
3 GPON ONU Connections displayed

```

Creating a CPE IP common profile for VoIP (Optional)

The default CPE IP common profile specified the DHCP as the host IP option. It indicates CPE will get the host IP address automatically from the DHCP server.



Note: CPE IP common profile can only be deleted when it is not associated by any CPE IP profiles.

Command:

```

cpe <voip | pwe> ip-com add <profile-name>
    [ host-ip-option < dhcp | static > ]
    [ netmask < value > ]
    [ gateway < IP address > ]
    [ primary-dns < IP address > ]
    [ secondary-dns < IP address > ]

```

Table 71: cpe ip-com add Command Option Explanations

Command Option	Description
<i>profilename</i>	Specifies a unique CPE IP common profile.
host-ip-option <dhcp static >	Selects an IP related option. DHCP or static. DHCP is the default value. It indicates CPE will get the host IP address automatically from the DHCP server.
netmask <value>	Specifies the subnet mask for IP host services.
gateway <IP address>	Specifies the default gateway address used for IP host services, this attribute has default value 0.0.0.0.
primary-dns <IP address>	Specifies the primary DNS IP address. If this value is 0.0.0.0, no primary SIP DNS is defined. The default value is 0.0.0.0.
secondary-dns <IP address>	Specifies the secondary DNS IP address. If this value is 0.0.0.0, no second SIP DNS is defined. The default value is 0.0.0.0.

The following example creates a static CPE IP common profile for voice service:

- 1 Create a CPE IP common profile with profile-name. The profile index will be generated automatically.

```
zSH> CPE> VOIP> IP> IP-COM> add IPserver host-ip-option static netmask
255.255.255.0 gateway 172.168.3.254 primary-dns 172.168.19.1
Profile "IPserver" has been created with index 2
```

- 2 Show the default CPE IP common and user-created CPE IP commons.

```
zSH> CPE> VOIP> IP> IP-COM> show all
```

Index	Profile Name	host IP option	gateway	igmp fn	nat	secure fwd	default iface	dns src	dns type
1	Default_Cpe_Ip_Server	dhcp	0.0.0.0	none	nat	disabl	false	false	default
100001	default-lan-ip-server100001	static	0.0.0.0	none	disabl	disabl	false	false	default
2	IPserver	static	172.168.3.254	none	nat	disabl	false	false	default

3 entries found.

- 3 If users want to delete a user-created CPE IP common, use the delete command with the profile index or profile name.

```
zSH> CPE> IP> IP-COM> delete IPserver
Profile has been deleted.
```

Creating a CPE IP profile for the VoIP service and associate it with a CPE IP common profile

Create a CPE IP profile for the VoIP service and associate it with a CPE IP common profile. If there is no CPE IP common profile specified, the default CPE IP common profile will be used.

Command:

```
cpe voip ip add <interface>
[ host-ip <IP address > ]
[ ip-com <index | profile-name > ]
```

Table 72: cpe ip add Command Option Explanations

Command Option	Description
<i>interface</i>	Specifies an ONU interface ID.
host-ip <IP address>	Specifies the address used for IP host services. The default value is 0.0.0.0.
ip-com <index profile-name>	Associates a CPE IP common profile with this host IP. If this field is not specified or is 1, the default CPE IP common profile (index 1) with DHCP enabled will be used.

- 1 Create a CPE IP profile for VoIP service.

```
zSH> CPE> VOIP> IP> add 1/3/1
Service has been created
```

The default CPE IP common with index 1 will be used.

2 Show CPE IP profiles.

```
zSH> CPE> VOIP> IP> show all
      CPE      Service      host IP      IP Com Profile
=====
      1/3/1 VOIP      0.0.0.0      1
1 services displayed
```

3 Delete a CPE IP profile.

```
zSH> CPE> VOIP> IP> delete 1/3/1
Service has been deleted.
```

Creating CPE VoIP server profiles

This section will create a CPE VoIP server profile. Each CPE VoIP server profile specifies the IP addresses or names of the primary and secondary VoIP servers, and the VoIP signalling protocol. The VoIP signalling protocol could be SIP, SIP PLAR, H.248, or MGCP (available in RG only).



Note: CPE VoIP server profile can only be deleted when it is not associated by any CPE VoIP subscriber profiles.

Command:

```
cpe voip server add <profile-name>
[primary-server < 64 byte character string > ]
[secondary-server < 64 byte character string > ]
[signalling-protocol < sip | siplar | h248 | mgcp > ]
[sip-domain < 64 byte character string > ]
[sip-registrar < 64 byte character string > ]
[mgc-termination-id-base < 25 byte character string > ]
[oob-dtmf-events < enabled | disabled > ]
[oob-cas-events < enabled | disabled > ]
[softswitch < 4 byte vendor code > ]
[outbound-server < 255 byte character string > ]
[port-id < 2 bytes, default -1 > ]
[dtmf-events-passing-method < rfc4733 | sipinfo > ]
[cas-events-passing-method < rfc4733 | sipinfo > ]
[rtp-dscp < 0-63 | af11 | af12 | af13 | af21 | af22
| af23 | af31 | af32 | af33 | af41 | af42 | af43 |
cs1 | cs2 | cs3 | cs4 | cs5 | cs6 | cs7 | default |
ef > ]
```

```

[signalling-dscp < 0-63 | af11 | af12 | af13 | af21
| af22 | af23 | af31 | af32 | af33 | af41 | af42 |
af43 | cs1 | cs2 | cs3 | cs4 | cs5 | cs6 | cs7 |
default | ef > ]

[sip-reg-exp-time ]

[sip-rereg-head-start-time ]

[sip-reg-retry-time ]

[release-timer]

[roh-timer]

mgcp-client-address-mode <ip | ipbracketed |
domainname>

mgcp-persistent-notify<enabled | disabled>

```

[Table 73](#) provides the description for command options in the **cpe voip server add** command.

Table 73: cpe voip server add Command Option Explanations

Command Option	Description
<i>profilename</i>	Specifies a unique CPE VoIP server profile.
primary-server value	Contains the name (IP address or resolved name) of the primary MGC or SIP proxy server that controls the signalling messages.
secondary-server value	Contains the name (IP address or resolved name) of the secondary or backup MGC proxy server that controls the signalling messages.
signalling-protocol < sip siplar h248 mgcp >	Specifies the VoIP signalling protocol. By default, it is h248.
sip-domain < 64 byte character string >	Contains the host or domain part of the SIP address of record for users connected to this ONT.
sip-registrar < 64 byte character string >	Contains the name (IP address or resolved name) of the registrar server for SIP signalling messages.
mgc-termination-id-base < 25 byte character string >	Specifies the base string for the H.248 physical termination id's for this ONT. This string is intended to uniquely identify an ONT. Vendor specific termination identifiers are optionally added to this string to uniquely identify a termination on a specific ONT.
oob-dtmf-events < <i>enabled</i> <i>disabled</i> >	When the oob-dtmf-events is enabled, Dual-Tone Multi-Frequency (DTMF) signals are carried out of band via RTP or the associated signalling protocol. When disabled, DTMF tones are carried in the PCM stream, and the settings of the dtmf-events-passing-method is ignore. Disabled is the default value.

Table 73: cpe voip server add Command Option Explanations

Command Option	Description
oob-cas-events < enabled disabled >	When the oob-cas-events is enabled, Dual-Tone Multi-Frequency (DTMF) signals are carried out of band via RTP or the associated signalling protocol. When disabled, DTMF tones are carried in the PCM stream, and the settings of the dtmf-events-passing-method is ignore. Disabled is the default value.
softswitch < 4 byte vendor code >	Defines the SIP gateway SoftSwitch vendor. The format is four ASCII coded alphabetic characters[A..Z]. By default, metaswitch is used. Here is the list of SoftSwitch 4-character codes that supported for the zNID 24xx. (The other ONT models might support different codes, refer to the <i>ONU configuration guide</i> for the details.) AX2K: Axtel CS2K BSFT: Broadsoft CRPK: Cirpack CCOM : CopperCom ERIC: Ericsson GBND: GenBand G6 HWEI: Huawei SoftX3000 META: Metaswitch NSHQ: Nokia Siemens HiQ NTEL: Nortel CS1500 / GenBandC15, Nortel CS2K / GenBandCS20 NTWK: Network Only OSER: OpenSer TRUA: Taqua T7000 UTSI: UT StarCom URAL: Huawei IMS VXTL: VixTel
outbound-server < 4255 byte character string>	Contains the name (IP address or resolved name) of the outbound proxy server for SIP signalling messages.
port-id < 2 bytes, default -1>	This attribute specifies the TCP/UDP port number of the VoIP protocol. The default value -1 selects the default port number for the VoIP protocol. It is 2944 for H.248 and 5060 for SIP.
dtmf-events-passing-method < rfc4733 sipinfo>	If dtmf-events-passing-method = rfc4733, DTMF digits are carried in the RTP payload. rfc4733 is the default value. If dtmf-events-passing-method = sipinfo, DTMF digits are carried along the signalling path in the INFO messages. Note that SIPINFO is only available in the 24xx ONUs.
cas-events-passing-method < rfc4733 sipinfo>	If cas-events-passing-method = rfc4733, Channel-Associated Signaling (CAS) digits are carried in the RTP payload. rfc4733 is default value. If cas-events-passing-method = sipinfo, CAS digits are carried along the signalling path in the INFO messages. Note that SIPINFO is only available in the 24xx ONUs.

Table 73: cpe voip server add Command Option Explanations

Command Option	Description
rtp-dscp <0-63 af11 af12 af13 af21 af22 af23 af31 af32 af41 af42 af43 cs1 cs2 cs3 cs4 cs5 cs6 cs7 default ef>	Set the Differentiated services codepoint value for RTP streams associated with the VoIP server.
signalling-dscp <0-63 af11 af12 af13 af21 af22 af23 af31 af32 af41 af42 af43 cs1 cs2 cs3 cs4 cs5 cs6 cs7 default ef>	Set the DSCP (Differentiated Services Code Point) value for signalling messages associated with the VoIP server. The value of the DSCP is used to prioritize traffic through the network.
sip-reg-exp-time	Specifies the SIP registration expiration time in seconds. If it is 0, the SIP agent does not add an expiration time to the registration requests and does not perform re-registration. Default is 3600.
sip-rereg-head-start-time	Specifies the time in seconds prior to timeout that causes the SIP agent to start the re-registration process. Default is 360. The recommended value for Zhone Broadcom-based zNIDs is 15.
sip-reg-retry-time	Specifies the SIP registration retry time in seconds. Default is 60.
release-timer	Release timer in seconds. The value 0 specifies that the ONT is to use its internal default. Default is 10.
roh-timer	This attribute defines the time in seconds for the receiver off hook condition before ROH tone is applied. The value 0 disables ROH timing. Default value is 15.
mgcp-client-address-mode <ip ipbracketed domainname>	IP and IPbracketed will cause the MGCP client name to be the bound voice host IP address. Domainname will allow the users to input any user text string used in accessing the call agent, usually a domain name. Most customers will use IP or IPbracketed mode. The default value is IP.
mgcp-persistent-notify <enabled disabled>	When enabled, all switchhook events will be forwarded to the switch immediately without regards to what the switch has requested. When disabled, only the events that the switch has requested will be forwarded.

To create VoIP server profiles, use the **cpe voip server add** command.

- 1 This example creates a VoIP server profile for a SIP VoIP server.

```
zSH> CPE> VOIP> SERVER> add metaswitch-sip primary-server 172.16.60.51
signalling-protocol sip sip-domain metaswitch.oak.zhone.com sip-registrar
metaswitch.oak.zhone.com
Profile "metaswitch-sip" has been created with index 1
```

- 2 Enable the oob-dtmf-events, and set the dtmf-events-passing-method to rfc4733 (default value).

```
zSH> CPE> VOIP> SERVER> modify 1 oob-dtmf-events enabled
dtmf-events-passing-method rfc4733
Profile has been modified.
```

- 3 Enable the oob-cas-events, and set the cas-events-passing-method to rfc4733 (default value).

```
zSH> CPE> VOIP> SERVER> modify 1 oob-cas-events enabled cas-events-passing-method rfc4733
```

Profile has been modified.

- 4 Shows the VoIP server profile.

```
zSH> CPE> VOIP> SERVER> show metaswitch-sip
```

Index	Profile Name	Signalling Protocol	Oob Events	Dtmf Events	Oob Events	Cas Events	Dtmf Events Passing Method	Cas Events Passing Method
1	metaswitch-sip	sip	disabled	disabled			rfc4733	rfc4733
	Primary Server	: 172.16.60.51						
	Secondary Server	: 0.0.0.0						
	Sip Domain	: metaswitch.oak.zhone.com						
	Sip Registrar	: metaswitch.oak.zhone.com						
	Mgc Termination Id Base	:						
	Softswitch	:						
	OutBound Server	:						
	Port Id	: -1						
	Rtp Dscp	: 0						
	Signalling Dscp	: 0						
	Sip Reg Exp Time	: 3600						
	Rereg Head Start Time	: 360						
	Sip Reg Retry Time	: 60						
	Release Timer	: 10						
	Roh Timer	: 15						
	MGCP Address Mode	: ip						
	MGCP Persistent Notify	: disabled						

1 entries found.

Creating CPE SIP dial plans for a SIP VoIP server (optional)

CPE SIP dialplans are only for SIP. Users can create up to 30 CPE SIP dialplans for each CPE SIP VoIP server.

Command:

```
cpe voip dialplan add < server-index | server-profile-name >
[dial-plan-format < h248 | nsc | vendor-specific > ]
[dial-plan < 25 character string > ]
```

Table 74: cpe sip dialplan add Command Option Explanations

Command Option	Description
<i>profilename</i>	Specifies a unique CPE VoIP server profile name.
dial-plan-format <h248 nsc vendor-specific >	Defines the dialplan format standard that is supported on the ONT for VoIP service. It could be h248, nsc, or vendor-specific. The default value is h248.
dial-plan < 25 character string >	Defines the dialplan used by the VoIP service.

- 1 Create the first CPE SIP dialplan profile for SIP VoIP server 1.

The vertical bar in this example are entered by pressing Shift and backslash keys together.

```
zSH> CPE> VOIP> DIALPLAN> add 1 dial-plan 1xx|[2-7]xxxxxx
Profile has been created with index 1/1
```

- 2 Create the second CPE SIP dialplan profile for the same SIP VoIP server 1.

```
zSH> CPE> VOIP> DIALPLAN> add 1 dial-plan xx.T|*xx.T
Profile has been created with index 1/2
```

- 3 Show all the SIP dialplans.

```
zSH> CPE> VOIP> DIALPLAN> show all
      index      dial-plan-format      dial-plan
=====
      1/1          h248          1xx|[2-7]xxxxxx
      1/2          h248          xx.T|*xx.T
2 entries found.
```

Creating CPE VoIP features profile for SIP or SIP PLAR

Create a CPE VoIP features profile for the VoIP service. The VoIP features profiles are using the bitmaps format.



Note: The CPE VoIP features profile is only applicable for SIP or SIP PLAR VoIP server.



Note: CPE VoIP feature profile can only be deleted when it is not associated by any CPE VoIP subscriber profiles.

Command:

```
cpe voip features add <profile-name>

[announcement-type < silence | reordertone |
fastbusy | voice | na > ]

[ cid-features < calling-number | calling-name |
cid-block | cid-number | cid-name | anonym-block |
all | none > ]

[ call-waiting-features < calling-waiting |
cid-announcement | all | none > ]

[ call-progress-or-transfer-features < 3-way |
call-transfer | call-hold | call-park |
do-not-disturb | flash-on-emergency |
emergency-hold | 6-way | all | none> ]

[ call-presentation-features < msg-wait-splash-ring
| msg-wait-special-dial-tone | msg-wait-visual |
call-fwd | all | none> ]

[ hotline < disabled | hot | warm> ]

[hotline-number <PhoneNumber> ]
```


[warmline-timer <Timer>]

Table 75: cpe voip features add Command Option Explanations

Command Option	Description
<i>profilename</i>	Specifies a unique CPE voip features profile.
announcement-type < silence reordertone fastbusy voice na >	specifies the treatment when a subscriber goes off hook but does not attempt a call. Default: reordertone
cid-features < calling-number calling-name cid-blocking cid-number cid-name anonym-block all none >	Specifies the bit map of the caller ID features. Default: by default, all the bits are set
call-waiting-features < call-waiting cid-announcement all none >	Specifies the bit map of the call waiting features. Default: by default, all the bits are set
call-progress-or-transfer-features < 3-way call-transfer call-hold call-park do-not-disturb flash-on-emergency emergency-hold 6-way all none >	Specifies the bit map of the call processing features. Default: by default, all the bits are set
call-presentation-features < msg-wait-splash-ring msg-wait-special-dial-tone msg-wait-visual call-fwd all none >	Specifies the bit map of call presentation features. Default: by default, all the bits are set
hotline < disabled hot warm >	When the hotline is hot, the phone will immediately dial the hotline number. When the hotline is warm, the phone wait for the period specified in warmline-timer in ms before automatically dial the hotline number. Default: disabled
hotline-number < <i>PhoneNumber</i> >	The number this phone will automatically dial, if hotline or warmline feature is enabled.
warmline-timer < <i>Timer</i> >	The wait period before a warmline automatically dial the hotline number. The unit is milliseconds. Default: 200

- 1 Add a CPE VoIP features profile. By only specifying profile-name field, the profile will be created with all the default settings.

```
zSH> CPE> VOIP> FEATURES> add featureslist1
Profile "featurelist1" has been created with index 2
```

- 2 Show all the CPE VoIP features profiles, including the default and the user-created profiles.

```
zSH> CPE> VOIP> FEATURES> show all
cpe-voip-features 1      Name: Default_Cpe_Voip_Features      Type: reordertone
```

```

hotLine: warm                hotline-number: 7777000        warmline-timer: 3000
  Caller ID                  Call Waiting              Call Progress or      Call Presentation
    Features                  Features              Transfer Features      Features
=====
calling-number               call-waiting          3-way                 msg-wait-splash-ring
calling-name                 cid-announcement     call-transfer          msg-wait-special-dial-tone
cid-blocking                 call-hold             call-park              msg-wait-visual
cid-number                   do-not-disturb        flash-on-emergency     call-fwd
cid-name                     emergency-hold
anonym-block                 6-way

```

```

cpe-voip-features 2          Name: featurelist1          Type: reordertone
hotLine: disabled
  Caller ID                  Call Waiting              Call Progress or      Call Presentation
    Features                  Features              Transfer Features      Features
=====
calling-number               call-waiting          3-way                 msg-wait-splash-ring
calling-name                 cid-announcement     call-transfer          msg-wait-special-dial-tone
cid-blocking                 call-hold             call-park              msg-wait-visual
cid-number                   do-not-disturb        flash-on-emergency     call-fwd
cid-name                     emergency-hold
anonym-block                 6-way

```

2 entries found.

3 Modify the CPE VoIP features profile.

- To clear a bit map value, place a minus sign in front of the argument.
For example: “-calling-name” clears the calling-name value in the cid-features.
- To clear all features in a bit-map, use the “none” keyword.
- To enable all features in a bit-map, use the “all” keyword.

```

zSH> CPE> VOIP> FEATURES> modify featurelist1 cid-features -calling-name
call-waiting-features none call-progress-or-transfer-features all
call-presentation-features -call-fwd hotline warm hotline-number 7777437
warmline-timer 3000
Profile has been modified.

```

4 Show the CPE VoIP features profile.

```

zSH> CPE> VOIP> FEATURES> show featurelist1
cpe-voip-features 2          Name: featurelist1          Type: reordertone
hotLine: warm                hotline-number: 7777437        warmline-timer: 3000

  Caller ID                  Call Waiting              Call Progress or      Call Presentation
    Features                  Features              Transfer Features      Features
=====
calling-number               3-way                 msg-wait-splash-ring
cid-blocking                 call-transfer          msg-wait-special-dial-tone

```

cid-number	call-hold	msg-wait-visual
cid-name	call-park	
anonym-block	do-not-disturb	
	flash-on-emergency	
	emergency-hold	
	6-way	

1 entries found.

Creating CPE VoIP media profile

Create a CPE VoIP media profile for the VoIP service.



Note: CPE VoIP media profile can only be deleted when it is not associated by any CPE VoIP subscriber profiles.

Command:

```
cpe voip media add <profile-name>
[ echo-cancel < enabled | disabled > ]
[ fax-mode < pass-through | t38 > ]
[ codec-selection-first-order < pcmu | gsm | g723 | dvi4-8 |
      dvi4-16 | lpc | pcma | g722 | l16.2 | l16.1 |
      qcelp | cn | mpa | gy28 | dvi4-22 | g729 > ]
[ packet-period-selection-first-order < 10 .. 30 > ]
[ silence-suppression-first-order < enabled | disabled > ]
[ codec-selection-second-order < pcmu | gsm | g723 | dvi4-8 |
      dvi4-16 | lpc | pcma | g722 | l16.2 | l16.1 |
      qcelp | cn | mpa | gy28 | dvi4-22 | g729 > ]
[ packet-period-selection-second-order < 10 .. 30 > ]
[ silence-suppression-second-order < enabled | disabled > ]
[ codec-selection-third-order < pcmu | gsm | g723 | dvi4-8 |
      dvi4-16 | lpc | pcma | g722 | l16.2 | l16.1 |
      qcelp | cn | mpa | gy28 | dvi4-22 | g729 > ]
[ packet-period-selection-third-order < 10 .. 30 > ]
[ silence-suppression-third-order < enabled | disabled > ]
[ codec-selection-fourth-order < pcmu | gsm | g723 | dvi4-8 |
      dvi4-16 | lpc | pcma | g722 | l16.2 | l16.1 |
      qcelp | cn | mpa | gy28 | dvi4-22 | g729 > ]
[ packet-period-selection-fourth-order < 10 .. 30 > ]
[ silence-suppression-fourth-order < enabled | disabled > ]
```

This command creates a new profile. The <profile-name> must be supplied and must be unique for profile type. The profile index will be automatically generated.

Table 76: cpe voip media add Command Option Explanations

Command Option	Description
<i>profilename</i>	Specifies a unique CPE voip media profile.
echo-cancel < enabled disabled >	Turns on or off echo cancellation
fax-mode < pass-through t38 >	Selects the fax mode
codec-selection-n < pcmu gsm g723 dvi4-8 dvi4-16 lpc pcma g722 l16.2 l16.1 qcelp cn mpa gy28 dvi4-22 g729 >	Specifies the codec selection as defined by RFC 3551, n is in the range of “first-order” to “fourth-order”. The codec selection could be pcmu, gsm, g723, dvi4-8, dvi4-16, lpc, pcma, g722, l16.2, l16.1, qcelp, cn, mpa, gy28, dvi4-22, g729.
packet-period-selection-n < 10 .. 30 >	Packet period selection interval is Voice Sample Size in milliseconds. It specifies the time that the DSP will encode voice before sending. The longer the time the more propagation delay in the data stream, but also the more efficient the packetization. n is in the range of “first-order” to “fourth-order”.
silence-suppression-n < enabled disabled >	Specifies whether silence suppress is on or off. n is in the range of “first-order” to “fourth-order”.

1 Create a CPE VoIP media profile.

```
zSH> CPE> VOIP> MEDIA> add T38fax fax-mode t38
Profile "T38fax" has been created with index 2
```

2 Show CPE VoIP media profiles.

```
zSH> CPE> VOIP> MEDIA> show all
```

Index	Profile Name	echo cancel	fax mode	codec Selection	Packet-period selection	silence suppression
1	Default_Cpe_Voip_Media	enabled	passThrough	PCMU (1st)	10 (1st)	disabled (1st)
				PCMU (2nd)	10 (2nd)	disabled (2nd)
				PCMU (3rd)	10 (3rd)	disabled (3rd)
				PCMU (4th)	10 (4th)	disabled (4th)
2	T38fax	enabled	T38	PCMU (1st)	10 (1st)	disabled (1st)
				PCMU (2nd)	10 (2nd)	disabled (2nd)
				PCMU (3rd)	10 (3rd)	disabled (3rd)
				PCMU (4th)	10 (4th)	disabled (4th)

2 entries found.

Creating a CPE VoIP subscriber profile and associate it with a VoIP server, a VoIP features profile, and a media profile

To create a CPE VoIP subscriber profile on ONU POTS ports, use the **cpe voip add** command. When creating a CPE VoIP subscriber profile, user must specify a VoIP server profile, associate the VoIP server information to the POTS port. There is no default VoIP server profile. A VoIP features profile and a VoIP media profile are also required when creating the CPE VoIP

subscriber profile, if users do not specify these two profiles, then the default profiles are used.

Note that a CPE VoIP subscriber profile will also be created *automatically* if users set the ONU UNI POTS port's admin-state with the **gpononu port** command. For the details, refer to [Administration of subscriber facing ports on page 914](#).

After a CPE VoIP subscriber profile is created, to change the settings in that profile, users can use the **cpe voip modify** command, which has the same command syntax as the **cpe voip add** command.

Command:

```
cpe voip add <interface>[/<port number>]
    [ admin-state < up | down > ]
    [ dial-number < 36 byte character string > ]
    [ username < 25 byte unique character string > ]
    [ password < 32 byte character string > ]
    [ rx-gain < -12db - 6db > ]
    [ tx-gain < -12db - 6db > ]
    [ voip-server-profile <index | profile-name>]
    [ voip-features-profile <index | profile-name>]
    [ voip-media-profile <index | profile-name>]
    [ phone-follows-wan <enabled | disabled>]
```

Table 77: cpe voip add Command Option Explanations

Command Option	Description
<i>interface/port number</i>	ONU port ID and POTS UNI port ID of the physical interfaces.
admin-state < up down>	Activates or deactivates the functions performed by the POTS port for this subscriber. Default is up.
dial-number < 36 byte character string >	Text Field to specifies the subscriber directory number.  Note: The dial-number field must be specified in SIP and SIP PLAR configuration.
username < 25 byte unique character string >	Text Field that identifies the port to the switch. This must match what the Service Provider has set.  Note: The username field must be specified in SIP, SIP PLAR, and MGCP configuration.

Table 77: cpe voip add Command Option Explanations

Command Option	Description
password < 32 byte character string >	Contains the SIP user identification used for authentication.  Note: The password field must be specified in SIP, and it is only used with SIP.
rx-gain < -12db - 6db >	Specifies a gain value for the signal received from the network and sent toward the phone. Valid values are -12 (-12.0 dB) to 6 (+6.0 dB). A typical value of -7 will provide 7dB of total loss in the rx path.
tx-gain < -12db - 6db >	Specifies a gain value for the signal transmitted into the network from the phone. Valid values are -12 (-12dB) to 6 (+6dB). A typical value of -2 will provide 2 dB of total loss in the tx path.
voip-server-profile <index profile-name>	Associated cpe-voip-server profile.
voip-features-profile <index profile-name>	Associated cpe-voip-feature profile. If user specify profile index 1 or omit this field, a default profile is used.  Note: The voip-features-profile field is only used with SIP.
voip-media-profile <index profile-name>	Associated cpe-voip-media profile. If user specify profile index 1 or omit this field, a default profile is used.
phone-follows-wan <enabled disabled>	When enabled the phone will lose power any time the WAN is operation status of down. This will allow line monitoring equipment to detect loss of service.

To create a CPE VoIP subscriber profile on an ONU with the **cpe voip add** command:

- 1 Create SIP services on ONU 1/3/1 POTS 1, and associate VoIP server profile 1, the default VoIP features profile, and the default VoIP media profile with them.

Make sure the POTS port matches the port ID assigned during the creation of the subscriber facing MXK bridge and CPE connection.

```
zSH> CPE> VOIP> add 1/3/1/1 dial-number 2012020013
username 2012020013 password 123456
voip-server-profile 1
Service has been created
```

- 2 Create SIP services on ONU 1/3/1 POTS 2, and associate VoIP server profile 1, VoIP features profile featurelist1, and VoIP media profile T38fax with them.

Note that dial-number, username, and password are required for SIP configuration.

```

zSH> CPE> VOIP> add 1/3/1/2 dial-number 2012020014
username 2012020014 password 123456
voip-server-profile 1 voip-features-profile
featurelist1 voip-media-profile T38fax
Service has been created

```

3 Show all the VoIP subscriber profiles.

```

zSH> CPE> VOIP> show all

```

CPE	Port Number	Admin State	Rx Gain	Tx Gain	Voip-Server Profile Index	Voip Features Profile Index	Voip Media Profile Index
1/3/1	1	up	0	0	1	1	1
Dial Number : 2012020013							
Username : 2012020013							

CPE	Port Number	Admin State	Rx Gain	Tx Gain	Voip-Server Profile Index	Voip Features Profile Index	Voip Media Profile Index
1/3/1	2	up	0	0	1	2	2
Dial Number : 2012020014							
Username : 2012020014							

2 services displayed

Testing the VoIP configuration

- 1 Connect an ONT downlink POTS port to a VoIP phone.
Make sure the POTS port number matches the one users assigned for voice service with the **bridge add** command. In this example, users can connect either POTS 1 or POTS 2 to the VoIP phone.
- 2 Pick up the phone, users should be able to hear the dial tone and be able to make and receive a phone call.
This is the last step of the Voice configuration for SIP service in RG.
For the other voice signalling protocols configuration in RG, use the following procedures.

Create PWE on Dynamic OMCI on Uplink and Downlink Bridges

For Pseudo-Wire Emulation (PWE) service Zhong recommends using uplink and downlink bridge configurations.

To create PWE service on the PWE (i.e. CES) ports on the same ONU, use the following steps:

- [Creating a GPON traffic profile, page 780](#)
- [Creating Uplink/Downlink bridge and CPE connection, page 780](#)
- [Creating CPE IP common profile for PWE, page 781](#)

- [Creating a CPE IP profile for the PWE service and associate it with a CPE IP common profile, page 782](#)
- [Creating a CPE PWE profile, page 782](#)
- [Creating PWE service on ONU CES ports and associate it with a CPE PWE profile, page 785](#)
- [Testing the PWE configuration, page 786](#)

Creating a GPON traffic profile

Add the GPON traffic profile.

The following GPON traffic profile is recommended for up to four PWE (i.e. CES) ports. Note that it is not recommended to use the DBA configuration fields in the GPON traffic profile for PWE service.

```
zSH> new gpon-traffic-profile 4

gpon-traffic-profile 4
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}: 13312
traffic-class: -----> {ubr}: cbr
compensated: -----> {false}:
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: ----> {0}:
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved..
```

Creating Uplink/Downlink bridge and CPE connection

This section will create an uplink bridge and a downlink bridge on the MXK and create CPE connection on the ONU for PWE service:

- 1 Create an uplink bridge on the uplink interface.

```
zSH> bridge add 1-a-4-0/eth uplink vlan 503 tagged
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-503/bridge
Bridge-path added successfully
```

- 2 Create a downlink bridge on the downlink interface and create a CPE connection for all the CES ports on the ONU.

This example creates a PWE service for all CES ports on the ONU 1/3/1 and inserts VLAN 503 with COS 3 to the tagged PWE packets.

```
zSH> bridge add 1-1-3-1/gpononu gem 1350 gtp 4 downlink cos 3 vlan 503 tagged pwe
Adding bridge on 1-1-3-1/gpononu
```


Created bridge-interface-record 1-1-3-1350-gponport-503/bridge
CPE Connection 1-1-3-1350/gponport/16/0/0/0 has been created

3 On MXK, run the **bridge show** command to view the MXK bridges.

zSH> **bridge show**

Orig						
Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn		Tagged 300	1/1/3/1/gpononu	1-1-3-710-gponport-300/bridge	UP	D 00:19:c7:0d:11:bf
dwn		Tagged 503	1/1/3/1/gpononu	1-1-3-1350-gponport-503/bridge	UP	
dwn		Tagged 999	1/1/3/1/gpononu	1-1-3-650-gponport-999/bridge	UP	
dwn		Tagged 1001	1/1/3/1/gpononu	1-1-3-610-gponport-1001/bridge	UP	
upl		Tagged 300	1/a/4/0/eth	ethernet4-300/bridge	UP	
upl		Tagged 503	1/a/4/0/eth	ethernet4-503/bridge	UP	
upl		Tagged 999	1/a/4/0/eth	ethernet4-999/bridge	UP	S VLAN 999 default
upl		Tagged 1001	1/a/8/0/eth	ethernet8/bridge	UP	S VLAN 1001 default

8 Bridge Interfaces displayed

4 View the CPE connections.

zSH> **bridge show onu**

ONU	GEM Port	ONU UNI	DSCP to COS	ONU UNI VLAN/SLAN	OLT VLAN/SLAN	OLT MVR	Service	OLT Bridge	ST
1/3/1	610	eth 1		1001/----	Tagged 1001		data	1-1-3-610-gponport-1001/bridge	UP
1/3/1	650	eth 3			Tagged 999		iptv	1-1-3-650-gponport-999/bridge	UP
1/3/1	710	pots			Tagged 300		sip	1-1-3-710-gponport-300/bridge	UP
1/3/1	1350	ces			Tagged 503		pwe	1-1-3-1350-gponport-503/bridge	UP

4 Bridge Interfaces displayed
4 GPON ONU Connections displayed

Creating CPE IP common profile for PWE

The default CPE IP common profile specified the DHCP as the host IP option. It indicates CPE will get the host IP address automatically from the DHCP server.



Note: CPE IP common profile can only be deleted when it is not associated by any CPE IP profiles.

The following example creates a static CPE IP common profile:

- 1 Create a CPE IP common profile with profile-name. The profile index will be generated automatically.

```
zSH> CPE> PWE> IP> IP-COM> add ip-pwe-server host-ip-option static netmask
255.255.255.0 gateway 172.10.10.1 primary-dns 172.111.142.50 secondary-dns
172.112.142.50
```

Profile "ip-pwe-server" has been created with index 2

- 2 Show the default CPE IP common profile and user-created CPE IP common profile.

zSH> CPE> PWE> IP> IP-COM> **show all**

Index	Profile Name	host IP option	gateway	igmp fn	nat	secure fwd	default iface	dns src	dns type
1	Default_Cpe_Ip_Server	dhcp	0.0.0.0	none	nat	disabl	false	false	default
2	IPserver	static	172.10.10.1	none	nat	disabl	false	false	default

2 entries found.

- 3 If users want to delete a user-created CPE IP common, use the delete command with the profile index or profile name.

```
zSH> CPE> IP> IP-COM> delete IPserver
Profile has been deleted.
```

Creating a CPE IP profile for the PWE service and associate it with a CPE IP common profile

Create a CPE IP profile for the PWE service and associate it with a CPE IP common profile. If there is no CPE IP common profile ID specified, the default CPE IP common profile with dhcp option will be used.

- 1 Create a CPE IP profile on ONU 1/3/1 for the PWE service, and associate it with an IP common profile. Note that the host-ip field must be specified if a static IP server is used.

```
zSH> CPE> PWE> IP> add 1/3/1 pwe host-ip 172.10.10.20 ip-com ip-pwe-server
Service has been created
```

- 2 Show all the CPE IP profiles.

```
zSH> CPE> PWE> IP> show all
CPE      Service      host IP      IP Com Profile
=====
1/3/1    VOIP            0.0.0.0      1
1/3/1    PWE            172.10.10.20 2
2 services displayed
```

Creating a CPE PWE profile

Create a CPE PWE profile for the common PWE service or use the default CPE PWE profile.



Note: CPE PWE profile can only be deleted when it is not associated by any CPE PWE subscriber profiles.

Command:

```
cpe pwe common add <profile-name>
[ line-type < ds1 | e1 > ]
[ encoding < b8zs | ami | hdb3 | b3zs > ]
[ timing-mode < network | differential |
adaptive | loop > ]
[ payload-size < value > ]
[ jitter-buf-max < value > ]
[ jitter-buf-desired < value > ]
[ DSCP < value > ]
```

Table 78: cpe pwe common add Command Option Explanations

Command Option	Description
<i>profile-name</i>	Specifies a unique profile name for cpe pwe profile.
line-type < ds1 e1 >	Specifies the line type used. ds1 or e1. Default value is e1.
encoding < b8zs ami hdb3 b3zs >	Specifies the line coding scheme. b8zs is used for ds1 line-type, hdb3 is used for e1 line-type. Default value is hdb3.
timing-mode < network differential adaptive loop >	Selects the timing mode of the TDM service. If RTP is used. Default value is network.
payload-size < <i>value</i> >	Specifies the number of payload bytes per packets. Valid only if service-type is unstructured or octetalignedunstruct (unstructured octet aligned). Valid choices depend on the TDM service, but must include the following. Other choices are at the vendor's discretion. Values: 192 For DS1 service 200 For DS1 service, required only if service-type octetalignedunstruct is selected 256 For E1 service 1024 For DS3 and E3 service.
jitter-buf-max < <i>value</i> >	Specifies the desired maximum depth of the playout buffer in the PSN to TDM direction. The value is expressed as a multiple of the 125 microseconds frame rate. The value 0 selects the ONT's internal policy.
jitter-buf-desired < <i>value</i> >	Specifies the desired nominal fill depth of the playout buffer in the PSN to TDM direction. The value is expressed as a multiple of the 125 microseconds frame rate. The value 0 selects the ONT's internal policy.

Table 78: cpe pwe common add Command Option Explanations

Command Option	Description
dscp < value >	Set the Differentiated services codepoint value for cpe-pwe. Values: 0-63 af11 af12 af13 af21 af22 af23 af31 af32 af33 af41 af42 af43 cs1 cs2 cs3 cs4 cs5 cs6 cs7 default ef

- 1
- Add a PWE profile. If the user-end device is PWE T1 device, line-type must be ds1, and encoding might be changed as well.

```
zSH> CPE> PWE> COMMON> add pwe-t1 line-type ds1 encoding b8zs
Profile "pwe-t1" has been created with index 2
```

If the user-end device is PWE e1 device, users can use the default value of the line-type and encoding, which are e1 and hdb3.

```
zSH> CPE> PWE> COMMON> add pwe-e1
```

- 2
- Show all the PWE profiles.

```
zSH> CPE> PWE> COMMON> show all

Index      Profile Name      line
=====  =====
1      Default_Cpe_Pwe  type
2      pwe-t1           encoding
Service type  timing Mode  payload jitter JBuf
=====  =====
1      Default_Cpe_Pwe  ds1   hdb3   unstructured adaptive 250 128 64
2      pwe-t1           ds1   b8zs   unstructured adaptive 250 128 64
2 entries found.
```

Creating PWE service on ONU CES ports and associate it with a CPE PWE profile

Create a CPE PWE subscriber profile on ONU CES ports *manually* and associate it with a CPE PWE profile by using the **cpe pwe add** command. If there is no CPE PWE profile specified, the default CPE PWE profile will be used.

Note that a CPE PWE subscriber profile will also be created *automatically* if users set the ONU UNI CES port's admin-state with the `gpononu port` command. For the details, refer to [Administration of subscriber facing ports on page 914](#).

After a CPE PWE subscriber profile is created, to change the settings in that profile, users can use the **cpe pwe modify** command, which has the same command syntax as the **cpe pwe add** command.

Command:

```
cpe pwe add <interface>/<port number>

    [ admin-state < up | down > ]
    [ near-end-port < value > ]
    [ far-end-ip < IP Address > ]
    [ far-end-port < value > ]
    [ line-length < value > ]
    [ pwe-profile < index | profile-name > ]
    [ line-status-alarm < enabled | disabled > ]
    [ alarm-severity < critical | major | minor | warning > ]
```

Table 79: cpe pwe add Command Option Explanations

Command Option	Description
<i>interface/port number</i>	ONU port ID and CES UNI port ID of the physical interfaces.
admin-state <i>value</i>	Activates or deactivates the functions performed by the CES port for this subscriber. Possible values are up, down. Default value is up.
near-end-port <i>value</i>	When the pseudowire service is transported via IP, this attribute specifies the port number of the near-end TCP/UDP service. Default is 57000 + port number.
far-end-ip <i>value</i>	When the pseudowire service is transported via IP, this attribute specifies the IP address or resolved name of the far-end termination point.
far-end-port <i>value</i>	When the pseudowire service is transported via IP, this attribute specifies the port number of the far-end TCP/UDP service. Default is 57000 + port number.
line-length <i>value</i>	Specifies the length of the twisted pair cable from a DS1 physical UNI to the DSX-1 cross-connect point. In the unit of feet. Default is 0.

Table 79: cpe pwe add Command Option Explanations

Command Option	Description
pwe-profile <i>index profile-name</i>	Points to the associated CPE PWE profile. If this field is not specified or is 1, the default CPE PWE profile (index 1) will be used.
line-status-alarm value	Enables or disables line status alarms on this port. Possible values are enabled or disabled. Default is enabled.
alarm-severity value	Sets the severity of line status alarms on this port. The severity level takes effect only after line-status-alarm has been enabled. Possible values are critical, major, minor, warning. Default value is major.

To create a CPE PWE subscriber profile with the **cpe pwe add** command:

- 1 Create a CPE PWE subscriber profile on ONU 1/3/1 CES port 1 and associate with a CPE PWE profile.

Make sure the CES port matches the port ID assigned during the creation of the subscriber facing MXK bridge and CPE connection.

```
zSH> CPE> PWE> add 1/3/1/1 near-end-port 57001 far-end-ip 10.10.10.1 far-end-port
57001 pwe-profile pwe-t1
Service has been created
```

- 2 Create another CPE PWE subscriber profile on ONU 1/3/1 CES port 2 and associate with a CPE PWE profile.

```
zSH> CPE> PWE> add 1/3/1/2 near-end-port 57002 far-end-ip 10.10.10.1 far-end-port
57002 pwe-profile pwe-t1
Service has been created
```

- 3 Show all the CPE PWE subscriber profiles

```
zSH> CPE> PWE> show 1/3/1
```

CPE	Port #	Admin	Near Port	Far End Ip	Far Port	Line Len	Profile	Alm St	Sev
1/3/1	1	up	57001	10.10.10.1	57001	0	2	En	Mj
1/3/1	2	up	57002	10.10.10.1	57002	0	2	En	Mj

2 services displayed

Testing the PWE configuration

- 1 Connect an ONT downlink CES port to a T1/E1 device.

Make sure the line type of the device matches the configuration users specified. In this example, users can use a T1 device.

Make sure the CES port number matches the one users assigned for PWE service. In this example, users can connect CES 1 or CES 2 to the T1 device.

- 2 Verify if the T1/E1 device can get PWE service.

This test case are using a T1/E1 device has a Sync light. If the Sync light (green) is on, it means the T1/E1 device got service.

Create RF on Dynamic OMCI

Users are going to create RF service on another ONU with ONU model ZNID-GPON-2511. Same as ZNID-GPON-5114, before creating service on this ONU, users have to make sure the internal ME profile of this ONU is specified in the **onu set** command. RF service is different than other services, there is no need to create bridge configurations for it.

To create RF service on the RF ports on an ONU, use the following steps:

- [Specifying internal ME profile and activating the ONU, page 787](#)
- [Creating RF service on ONU RF ports, page 788](#)
- [Testing the RF configuration, page 789](#)

Specifying internal ME profile and activating the ONU

Based on the ONU model, assign the matching internal ME profile to this ONU. It is a way to indicate this ONU is provisioned by dynamic OMCI.

- 1 Specify an internal ME profile for an ONU.

After specifying internal ME profile zhong-2511 for ONU 1/3/2, dynamic OMCI supports are associated with the ONT.

```
zSH> onu set 1/3/2 meprof zhong-2511
```

- 2 To activate an ONT first run the **gpononu show** command to display the ONTs currently on the OLT, and discover the available serial numbers.

The **gpononu show** command has options to select by slot and OLT. If users run the command without defining the slot/OLT the command will check for ONTs on every port of every card and depending on the number of cards, may take a long time to complete.

```
zSH> gpononu show 1/3
```

```
Free ONUs for slot 1 olt 3:
```

2	3	4	5	6	7	8	9	10	11	12	
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64								

```
Discovered serial numbers for slot 1 olt 3:
```

sernoID	Vendor	Serial Number	Time Discovered
2	ZNTS	56725008	JUL 10 01:11:18 2014

- 3 Run the **gpononu set** command to associate a ONU port ID to a discovered ONT's serial number:

```
zSH> gpononu set 1/3/2 2
```

```
Onu 2 successfully enabled with serial number ZNTS
56725008
```

Creating RF service on ONU RF ports

To create a CPE RF subscriber profile on ONU RF ports *manually*, use the **cpe rf add** command.

Note that a CPE RF subscriber profile will be created *automatically* if users set the ONU RF port's admin-state with the gpononu port command. For the details, refer to [Administration of subscriber facing ports on page 914](#).

After a CPE RF subscriber profile is created, to change the settings in that profile, users can use the **cpe rf modify** command, which has the same command syntax as the **cpe rf add** command.

Command:

```
cpe rf add <interface>/<port number>
[ admin-state < up | down > ]
[ line-status-alarm < enabled | disabled> ]
[ alarm-severity < critical | major | minor | warning>
```

Table 80: cpe rf add Command Option Explanations

Command Option	Description
<i>interface/port number</i>	ONU port ID and Ethernet UNI port ID of the physical interfaces.
admin-state value	Activates or deactivates the functions performed by the RF port for this subscriber. Possible values are up, down. Default value is up.
line-status-alarm value	Enables or disables line status alarms on this port. Possible values are enabled or disabled. Default is disabled.
alarm-severity value	Sets the severity of line status alarms on this port. The severity level takes effect only after line-status-alarm has been enabled. Possible values are critical, major, minor, warning. Default value is major.

To create a CPE RF subscriber profile with the **cpe rf add** command:

- 1 Create an RF service on ONU 1/3/2 RF port 1. If there is no RF port provided then the next available port number will be chosen.

```
zSH> CPE> RF > add 1/3/2/1
```

- 2 Show all the RF subscriber profile.

```
zSH> CPE> RF> show all
  CPE      Port Number  Admin State  Alm St  Sev
=====
  1/3/2      1          up          Dis    Mj
1 services displayed
```

The default value of the Admin state is up.

- 3 Show the RF service on CPE level.


```

zSH> CPE> show 1/3/2
CPE 1/3/2
Service: RF
Port Admin
-----
      1      up

```

Testing the RF configuration

- 1 Connect an ONT downlink RF port to a TV through the RF cable.
Make sure the RF port number matches the one users assigned for RF service. In this example, users can connect RF 1 to the TV.
- 2 Check the TV signal.

Viewing all services on an ONU

To view all services on an ONU, use the **cpe show slot/olt/onu** command.

This example shows ONU 1/3/1 has configured with triple-play services and plus PWE service.

```

zSH> CPE> show 1/3/1

CPE 1/3/1
Service: DATA
GEM   UNI   UNI VLAN/SLAN OLT VLAN/SLAN Admin Oper   Rg-Mode
-----
  610  eth 1   1001/---- Tagged 1001   up   down

Service: IPTV
GEM   UNI   UNI VLAN/SLAN OLT VLAN/SLAN Admin Oper   Video Prof
-----
  650  eth 3           Tagged 6, 999   up   down           1

Service: SIP
GEM   UNI   UNI VLAN/SLAN OLT VLAN/SLAN   Host IP       IP Srvr Prof
-----
  710  pots           Tagged 300           1
      Port Admin Oper  Srvr Prof Feature Prof Media Prof DN       User Name
      Password
      =====
      =====
      1   up   up       1       1       1       2012020013       2012020013
123456
      2   up   up       1       2       2       2012020014       2012020014
123456

Service: PWE
GEM   UNI   UNI VLAN/SLAN OLT VLAN/SLAN   Host IP       IP Srvr Prof
-----
1350  ces           Tagged 503   172.10.10.20       2
      Port Admin Oper  Near End Port   Far End Ip     Far End Port Pwe Profile
      =====

```

1	up	down	57001	10.10.10.1	57001	2
2	up	down	57002	10.10.10.1	57002	2

Deletion of CPE profiles and CPE connection that associated on an ONU

Deleting CPE profiles and CPE connection that associated on an ONU

The **cpe delete** slot[/olt[/onu]] command deletes:

1. all the CPE subscriber profiles that were created on ONUs,
2. other CPE profiles that are associated on ONUs, such as CPE system profile if it was configured in RG provisioning,
3. CPE connections.

Note that if users want to delete all the ONU configuration on an ONU, use the **onu delete** slot[/olt[/onu]] command. For the details, refer to [Deleting ONU configuration on an ONU on page 917](#).

- 1 Show the CPE profiles and CPE connections on ONU 3/4/2.

```
zSH> cpe show 3/4/2
CPE 3/4/2
Service: DATA
GEM  UNI  UNI-VLAN/SLAN  VLAN/SLAN  G-VLAN Admin Oper  Rg-Mode
----
303  eth 1              Tagged 100      0    up   down B-Routed
Service: IPTV
GEM  UNI  UNI VLAN/SLAN OLT VLAN/SLAN G-VLAN Admin Oper  Video Prof Rg-Mode
----
403  eth 2              Tagged 200      0    up   down          Bridged
Service: SIP
GEM  UNI  UNI VLAN/SLAN OLT VLAN/SLAN G-VLAN  Host IP      IP Srvr Prof
----
503  pots              Tagged 300      0
cpe-system-common profile used: 1
```

- 2 Use **cpe delete** on ONU 3/4/2.

```
zSH> cpe delete 3/4/2
Ok to delete ALL CPE profiles for ONU 3/4/2? [yes] or [no]: yes
Do you want to exit from this request? [yes] or [no]: no
Are you sure? [yes] or [no]: yes
Operation completed
3 cpe-connection profiles deleted
1 Ethernet service deleted
1 SIP service deleted
1 Video service deleted
```

- 3 Verify the CPE subscriber profiles and CPE system profile are removed on this ONU.

```
zSH> CPE> ETH> show 3/4/2
No services found.
```

```
zSH> CPE> VOIP> show 3/4/2
No services found.
```

```
zSH> CPE> SYSTEM> show 3/4/2
No profiles found.
```

- 4 Verify all CPE connections and settings that were specified in the CPE subscriber profiles are removed on this ONU.

```
zSH> CPE show 3/4/2
No connections found for 1-3-4-2/gpononu
```

Residential Gateway (RG) Features Provisioning

There are GPON zNIDs where ONU and RG functions are physically integrated into the same device - these ONTs are referred to as Dual-Managed ONTs. Two Configuration Modes (i.e. ONT-Only Configuration Mode and ONT+RG Configuration Mode) exist to facilitate the provisioning of Dual-Managed ONTs under Unified Service Provisioning (USP) umbrella.

This section provides information on how to install and provision Dual-Managed GPON zNIDs with Unified Service Provisioning on the MXK.

- [RG Provisioning Overview on page 792](#)
- [CPE System Level Default Settings on page 827](#)
- [OMCI GPON zNID with RG Features Installation for Triple Services on page 799](#)
- [Static configuration on the WAN side interfaces \(without DHCP\) on page 834](#)
- [Static configuration on the LAN side interfaces with a new DHCP server on page 836](#)
- [Configuration of Static Routes on page 839](#)
- [Configuration of DNS Hosts and DNS Proxy on page 841](#)
- [Configuration of Firewall on page 844](#)
- [Configuration of DHCP Server on page 849](#)
- [Configuration of Conditional DHCP server on page 850](#)
- [Configuration of PPPoE username and password on page 853](#)
- [Configuration of TR-069 on page 855](#)
- [Set factory default for an ONU on page 856](#)
- [System Name and Location of zNID on page 857](#)
- [Guided VLAN on page 858](#)

- [PoE Power Control per Port & Total Power Budget on page 858](#)
- [Power Shedding Enable/Disable Per Port on page 859](#)

RG Provisioning Overview

This overview covers the following topics:

- [Configuration Modes on page 792](#)
- [Bridge add commands and RG modes in RG provisioning on page 792](#)

Configuration Modes

Dual-Managed GPON zNIDs may be provisioned via the ONT-Only Configuration Mode or by the ONT+RG Configuration Mode. With the ONT-Only mode, the zNID is managed via OMCI (as described in [Dynamic OMCI GPON zNID Installation on page 734](#)) and this is useful where services are limited to L2 Data and Video, or VoIP in simpler bridging cases. The vast majority of GPON zNIDs have been deployed in this configuration, with OMCI used exclusively for configuration and control. The provisioning requirements for this operating mode are defined by the ITU-T G.988 Standard.

However, provisioning RG capabilities on the Dual-Managed GPON zNIDs requires the use of the ONT+RG Configuration Mode. With this Mode an ONT+RG connection is managed through a combination of OMCI and SNMP. (Note that the users do not have to know what management protocols are used underneath when users provision the zNIDs.) Here, SNMP complements OMCI by being responsible for the configurations and management of the RG functions.

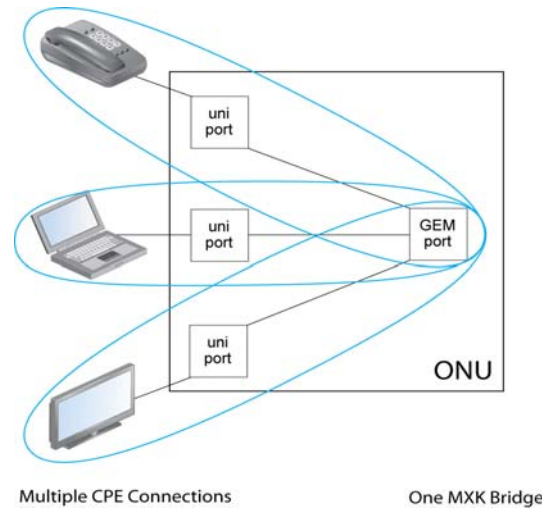
Both configuration modes allow for pre-provisioning, where the GPON zNIDs will be automatically configured once they are connected to the PON and come on-line. Note that the MXK currently only supports RG provisioning - status and statistics are retrieved either through OMCI or WebUI.

Bridge add commands and RG modes in RG provisioning

When using RG provisioning, MXK bridges and CPE connections can have one-to-one and one-to-many mappings.

The one-to-one mapping is shown in [Figure 117](#) in the Dynamic OMCI section. As shown in [Figure 119](#), the one-to-many mapping is one MXK bridge created on a GEM port that maps to multiple CPE connections created on multiple ONU UNI ports.

Figure 119: The one-to-many mapping between MXK bridges and CPE Connections



The MXK **“bridge add”** command with **“rg-operatingmode”** keyword creates a MXK bridge interface, and a ONT+ RG connection between a GEM port and the UNI(s).

The **“rg-operatingmode”** keyword specifies the operating mode of a VLAN in RG. The RG VLAN operating modes include:

- rg-bridged

The LAN-side interfaces and Wireless LAN interfaces can be a member of a bridged VLAN. A bridged VLAN can optionally have an IP address assigned to it for the purpose of enabling management, or supporting VoIP clients or PWE.

A RG bridge is created if **“rg-bridged”** mode is specified in the **“bridge add”** command. In the case of an ONU model only support ONU+RG configuration mode, does not support ONU-Only configuration mode(e.g. zNID 26xx, 42xx, 9xxx), a RG bridge is still created even without **“rg-bridged”** mode is explicitly specified in the command.

- rg-brouted

The rg-brouted mode operates like a bridged VLAN for all LAN-side interfaces, and as a routed VLAN for the WAN-side interface. In rg-brouted mode, there are only two IP interfaces: one for the routed WAN-side interface, and another for the bridged LAN-side interfaces.

A BRouted VLAN may have multiple LAN ports as members of the VLAN, and all ports will use the same IP subnet, and therefore the same DHCP server and IP server.

- rg-bpppoe

The PPPoE/Bridged VLANs are similar to Brouted VLANs, but the WAN interface is a PPPoE client that establishes a PPPoE tunnel to an upstream BRAS. On the LAN side of a PPPoE/Bridged VLAN, all ports will be members of the same IP Subnet.



Note: When a BRouted or PPPoE-Bridged VLAN is created via Unified Service Provisioning, the use-Derived-MAC option is automatically enabled on the zNID.

The “**bridge add**” command is a powerful command. If it is not a PPPoE connection, after the “**bridge add**” is executed, the zNid is ready to provide service. For PPPoE, the users have to set PPPoE user-ID and password too. (Refer to [Configuration of PPPoE username and password on page 853.](#))

[Table 81](#) provides a summary on how to add bridges for supported services in different RG operating modes.

Table 81: Creating services in RG by using bridge add command

Services	RG Operating Modes	Service Keywords	“Bridge Add” Command Examples
Data	rg-brouted (preferred mode for Data)	N/A Note: when no service keyword is specified, it implies data service.	<code>bridge add 1-4-1-1/gpononu gem 303 gtp 1 downlink vlan 100 tagged eth 1 rg-brouted</code>
	rg-bridged	N/A	<code>bridge add 1-4-1-1/gpononu gem 303 gtp 1 downlink vlan 100 tagged eth 1 rg-bridged</code>
	rg-bpppoe	N/A	<code>bridge add 1-4-1-1/gpononu gem 303 gtp 1 downlink vlan 100 tagged eth 1 rg-bpppoe</code>
Video	rg-bridged (preferred mode for Video)	video	<code>bridge add 1-4-1-1/gpononu gem 403 gtp 1 downlink vlan 200 tagged video 0/4 eth 2 rg-bridged</code>
	rg-brouted	video	<code>bridge add 1-4-1-1/gpononu gem 403 gtp 1 downlink vlan 200 tagged video 0/4 eth 2 rg-brouted</code>
	rg-bpppoe	video	<code>bridge add 1-4-1-1/gpononu gem 403 gtp 1 downlink vlan 200 tagged video 0/4 eth 2 rg-bpppoe</code>

Table 81: Creating services in RG by using bridge add command

Services	RG Operating Modes	Service Keywords	"Bridge Add" Command Examples
Voice	rg-bridged (preferred mode for Voice)	sip, sip-plar, mgcp	bridge add 1-4-1-1/gpononu gem 501 gtp 1 downlink vlan 300 tagged rg-bridged sip (It creates data path for SIP VoIP service on all POTS ports on the ONU) bridge add 1-4-1-1/gpononu gem 501 gtp 1 downlink vlan 300 tagged rg-bridged sipplar (It creates data path for SIP PLAR VoIP service on all POTS ports on the ONU) bridge add 1-4-1-1/gpononu gem 501 gtp 1 downlink vlan 300 tagged rg-bridged mgcp (It creates data path for MGCP VoIP service on all POTS ports on the ONU)
	rg-brouted	sip, sip-plar	bridge add 1-4-1-1/gpononu gem 501 gtp 1 downlink vlan 300 tagged rg-brouted sip (It creates data path for SIP VoIP service on all POTS ports on the ONU) bridge add 1-4-1-1/gpononu gem 501 gtp 1 downlink vlan 300 tagged rg-brouted sipplar (It creates data path for SIP PLAR VoIP service on all POTS ports on the ONU)
	rg-bpppoe	sip, sip-plar	bridge add 1-4-1-1/gpononu gem 501 gtp 1 downlink vlan 300 tagged rg-bpppoe sip (It creates data path for SIP VoIP service on all POTS ports on the ONU) bridge add 1-4-1-1/gpononu gem 501 gtp 1 downlink vlan 300 tagged rg-bpppoe sipplar (It creates data path for SIP PLAR VoIP service on all POTS ports on the ONU)
TR-069	rg-bridged (preferred mode for TR-069)	tr69	bridge add 1-4-1-1/gpononu gem 501 gtp 1 downlink vlan 300 tagged tr69 rg-bridged
	rg-brouted	tr69	bridge add 1-4-1-1/gpononu gem 501 gtp 1 downlink vlan 300 tagged tr69 rg-brouted
	rg-bpppoe	tr69	bridge add 1-4-1-1/gpononu gem 501 gtp 1 downlink vlan 300 tagged tr69 rg-bpppoe
PWE	rg-bridged	pwe	bridge add 1-4-1-1/gpononu gem 901 gtp 2 t1s vlan 60 tagged pwe rg-bridged

Creating multiple CPE connections that share one GEM port

The following example creates data service on a dual-managed ONT with RG-bridge mode. It creates one MXK bridge and two CPE connections. These two CPE connections share the same GEM port ID, VLAN, and GTP (GPON Traffic Profile).

1 Create a MXK bridge, and the first CPE connection:

```
zSH> bridge add 1-4-1-1/gpononu gem 303 gtp 1 downlink vlan 100 tagged eth 1
uni-vlan 100 rg-bridged
Adding bridge on 1-4-1-1/gpononu
Created bridge-interface-record 1-4-1-303-gponport-100/bridge
CPE Connection 1-4-1-303/gponport/1/1/100/0 has been created
```

The first part of this command “**bridge add 1-4-1-1/gpononu gem 303 gtp 1 downlink vlan 100 tagged**” creates a new MXK bridge. The second part of this command “**eth 1 uni-vlan 100 rg-bridged**” creates a ONT+RG connection between the GEM port and the UNI port. It also creates a bridged WAN-side interface and a bridged LAN-side interface in RG.

When specifying GEM ports in the “**bridge add**” command, note that any GEM port ID in the range of 257 to 3828 is allowed to be associated with any ONU except for GEM port 5xx, where the last two digits of the GEM port ID 5xx must be the ONU port number in the range from 1 to 64. For example, GEM port ID 564 must belongs to ONU 64. Each of these GEM port IDs needs to be unique for the OLT port.



Note: Some zNIDs models may reserve some GEM ports for different usage. Check with the *zNID Configuration Guide* to determine the available Unified Service Provisioning GEM port IDs.

GTP is a mandatory field in the **bridge add** command when creating a MXK bridge. It contains the bandwidth allocation information for the T-cont. For detail, refer to [Bandwidth Allocation for Upstream Traffic from the ONU to the MXK](#), page 925.

Show the MXK bridge:

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
dwn          100      1/2/1/0/vdsl  1-2-1-0-vdsl/bridge  DWN
dwn      Tagged 100  1/4/1/1/gpononu  1-4-1-303-gponport-100/bridge  UP  D 00:02:71:19:4b:28
                                         D 00:00:86:43:3c:e4
                                         D 192.168.1.12
dwn          200      1/6/1/0/eth   1-6-1-0-eth/bridge  UP
upl      Tagged 100  1/a/5/0/eth   ethernet5-100/bridge  UP  S VLAN 100 default
4 Bridge Interfaces displayed
```

Show CPE connections:

```
zSH> bridge show onu
GEM  ONU  DSCP  ONU UNI  OLT  OLT
ONU  Port  UNI   to COS  VLAN/SLAN  VLAN/SLAN  MVR  Service  Rg-Mode  OLT Bridge
ST
```



```

-----
4/1/1    303  eth 1          100/---- Tagged 100      data    Bridged 1-4-1-303-gponport-100/bridge
UP
1 Bridge Interfaces displayed
1 GPON ONU Connections displayed

```

Show WAN interfaces and LAN interfaces in RG:

```
zSH> cpe show 4/1/1
```

```
CPE 4/1/1
```

```
Service: DATA
```

GEM	UNI	UNI-VLAN/SLAN	VLAN/SLAN	Admin	Oper	Rg-Mode
303	eth 1	100/----	Tagged 100	up	up	Bridged

```
zSH> cpe rg wan show all
```

Ip-Com	Port-Fwd	Retry				Pppoe User Id
CPE	VLAN/SLAN	RG Mode	IP Address	Auth	Interval	
Profile	List	Profile				
4/1/1	100/----	Bridged	dhcp	--	--	--
1	0					

1 services displayed

```
zSH> cpe rg lan show all
```

CPE	UNI	UNI-Vlan/Slan	Vlan/Slan	IP-Address	IP Com Profile	Dhcp Srwr Profile
4/1/1	eth 1	100/----	Tagged 100	0.0.0.0	0	0

Bridged
Services displayed: 1

2 Create the second CPE connection, and map it to the newly created MXK bridge.

```
zSH> bridge add 1-4-1-1/gpononu gem 303 gtp 1 tm 1 downlink vlan 100 tagged eth 2 uni-vlan 100 rg-bridged
```

```
CPE Connection 1-4-1-303/gponport/1/2/100/0 has been created
```

The first part of this command “**bridge add 1-4-1-1/gpononu gem 303 gtp 1 tm 1 downlink vlan 100 tagged**” indicates the MXK bridge. The second part of this command “**eth 2 uni-vlan 100 rg-bridged**” creates ONT + RG connections with this MXK bridge. Since RG-Bridged VLAN has already been created by the previous command, this example adds Ethernet UNI port 2 as a member.

Show the MXK bridge:

```
zSH> bridge show
```

Orig	Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data

```

dwn          100      1/2/1/0/vdsl      1-2-1-0-vdsl/bridge      DWN
dwn          Tagged 100 1/4/1/1/gpononu      1-4-1-303-gponport-100/bridge  UP D 00:02:71:19:4b:28
                                         D 00:00:86:43:3c:e4
                                         D 192.168.1.12

dwn          200      1/6/1/0/eth      1-6-1-0-eth/bridge      UP
upl          Tagged 100 1/a/5/0/eth      ethernet5-100/bridge     UP S VLAN 100 default
4 Bridge Interfaces displayed

```

Show the CPE connections:

zSH> **bridge show onu**

```

      GEM  ONU  DSCP  ONU UNI  OLT  OLT
ONU   Port  UNI   to COS  VLAN/SLAN  VLAN/SLAN  MVR  Service  Rg-Mode  OLT Bridge
ST
-----
4/1/1  303  eth 1      100/---- Tagged 100      data   Bridged  1-4-1-303-gponport-100/bridge
UP
4/1/1  303  eth 2      100/---- Tagged 100      data   Bridged  1-4-1-303-gponport-100/bridge
DWN
1 Bridge Interfaces displayed
2 GPON ONU Connections displayed

```

Show the WAN interfaces and LAN interfaces in RG:

zSH> **cpe show 4/1/1**

CPE 4/1/1

Service: DATA

```

GEM  UNI  UNI-VLAN/SLAN  VLAN/SLAN  Admin Oper  Rg-Mode
----
303  eth 1      100/---- Tagged 100      up   up Bridged
303  eth 2      100/---- Tagged 100      up   down Bridged

```

zSH> **cpe rg wan show 4/1/1**

```

      CPE  VLAN/SLAN  RG Mode  IP Address  Auth  Retry  Ip-Com  Port-Fwd
      =====  =====  =====  =====  =====  =====  =====  =====
      4/1/1  100/----  Bridged      dhcp      --      --      1      0
Pppoe User Id: --
      4/1/1  100/----  Bridged      dhcp      --      --      1      0
Pppoe User Id: --
2 services displayed

```

zSH> **cpe rg lan show 4/1/1**

```

      CPE  UNI  UNI-Vlan/Slan  Vlan/Slan  IP-Address  IP Com  Dhcp Srvr
      Rg-Mode  Profile  Profile
      =====  =====  =====  =====  =====  =====  =====
      4/1/1 eth 1      100/---- Tagged 100      0.0.0.0      0      0
Bridged
      4/1/1 eth 2      100/---- Tagged 100      0.0.0.0      0      0
Bridged
Services displayed: 2

```

Deleting MXK bridge and associated CPE connections

The following example deletes a one-to-many mapping which has one MXK bridge mapping to two CPE connections.

To remove the MXK bridge and all the associated CPE connections.

- 1 To remove the MXK bridge and all the associated CPE connections at the same time, users can use this **bridge delete all** command

```
zSH> bridge delete 1-4-1-303-gponport-100/bridge all
CPE Connection 1-4-1-303/gponport/1/1/100/0 has been deleted
CPE Connection 1-4-1-303/gponport/1/2/100/0 has been deleted
1-4-1-303-gponport-100/bridge delete complete
```

- 2 Or users can remove the CPE connections first, and then remove the MXK bridge.

```
zSH> bridge delete 1-4-1-303-gponport-100/bridge eth 1 uni-vlan 100
CPE Connection 1-4-1-303/gponport/1/1/100/0 has been deleted
```

If there is only one CPE connection associated with the MXK bridge, users can delete the MXK bridge interface directly.

```
zSH> bridge delete 1-4-1-303-gponport-100/bridge
CPE Connection 1-4-1-303/gponport/1/2/100/0 has been deleted
1-4-1-303-gponport-100/bridge delete complete
```

OMCI GPON zNID with RG Features Installation for Triple Services

In this section, users will provision Data service, Video service, VoIP service on the same ONU, just the MXK bridge interface, GEM port setup, GPON traffic profile, VLAN, UNI ports are different. For ease of discussion each of the applications is described separately in this section.

Generally these are the steps to follow to configure the MXK to be able to manage OMCI GPON zNIDs with RG features:

- [Creation of Data Services in RG, page 799](#)
- [Creation of Video Services in RG, page 803](#)
- [Creation of Voice Services in RG, page 805](#)
- [Creation of Data services on Wireless interfaces, page 814](#)
- [Create PWE on RG on TLS Bridges, page 827](#)

Creation of Data Services in RG

This section shows how to create data services on a zNID Ethernet Uni-port with rg-brouted mode.

- [Creating a GPON traffic profile, page 800](#)
- [Specifying internal ME profile for the ONU, page 800](#)

Only need to specify the internal ME profile once for each ONU.

- [Creating uplink/downlink MXK bridges, and CPE connections in RG-brouted mode for data services in RG, page 800](#)
- [Activating the ONU, page 802](#)

Only need to activate the ONU once.

- [Performing other necessary USP Data related configuration, page 803](#)

Creating a GPON traffic profile

Create a GPON traffic profile. The GPON traffic profile is required for the GPON bridge add command.

```
zSH> new gpon-traffic-profile 1
gpon-traffic-profile 1
Please provide the following: [q]uit.
    guaranteed-upstream-bw: -> {0}:
    traffic-class: -----> {ubr}:
compensated: -----> {false}:
shared: -----> {false}:
dba-enabled: -----> {false}:true
    dba-fixed-us-ubr-bw: ----> {0}:2048
    dba-fixed-us-cbr-bw: ----> {0}:
    dba-assured-us-bw: -----> {0}:
    dba-max-us-bw: -----> {0}:4096
    dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Specifying internal ME profile for the ONU

By specifying the ONU ME profile in the initial setup on an ONU, the MXK knows this ONU model is provisioned by RG.

Users only need to specify the ME profile once.

```
zSH> onu set 4/1/1 meprof zhong-2426
```

Creating uplink/downlink MXK bridges, and CPE connections in RG-brouted mode for data services in RG

In the example below, Ethernet Uni-port 1 in RG is member of the Brouted VLAN 100. NAT is enabled on the WAN side. When creating the LAN-side interfaces in the same RG VLAN, one DHCP server and IP common server will be assigned to all interfaces. By default, the first assigned index number is 100001, and the IP subnet is 192.168.1.1.

The DHCP server index and IP server index number will be increased automatically if users created new CPE connection in another RG VLAN.

Users can use the default DHCP server and IP server or create and assign another DHCP server and IP common server to the LAN-side interface as users desired. For the details, refer to [Configuration of DHCP Server on page 849](#).

1 Create an uplink bridge interface on the MXK

```
zSH> bridge add 1-a-5-0/eth uplink vlan 100
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-100/bridge
bridge-path added successfully
```

2 Create a downlink MXK bridge, and a connection between the ONT 1-4-1-1 GEM port 303 and the CPE UNI Eth 1 on VLAN 100.

```
zSH> bridge add 1-4-1-1/gpononu gem 303 gtp 1 downlink
vlan 100 tagged eth 1 rg-brouted
Adding bridge on 1-4-1-1/gpononu
Created bridge-interface-record
1-4-1-303-gponport-100/bridge
CPE Connection 1-4-1-303/gponport/1/1/0/0 has been
created
```

The first part of this command, “**bridge add 1-4-1-1/gpononu gem 303 gtp 1 downlink vlan 100 tagged**” creates a new MXK bridge. The second part of this command, “**eth 1 rg-brouted**” creates a connection in CPE 1-4-1-1 to bridge untagged UNI port eth 1 to GEM port 303, a brouted WAN-side interface in RG, and a brouted LAN-side interface in RG.

3 View the bridges.

```
zSH> bridge show
```

Type	Orig VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
dwn		Tagged 100	1/4/1/1/gpononu	1-4-1-303-gponport-100/bridge	UP		
upl		Tagged 100	1/a/5/0/eth	ethernet5-100/bridge	UP	S	VLAN 100 default

2 Bridge Interfaces displayed

4 View the WAN interfaces.

```
zSH> CPE> RG> WAN> show 4/1/1
```

CPE	VLAN/SLAN	RG Mode	IP Address	Auth	Retry Interval	Ip-Com Profile	Port-Fwd List Profile	G-VLAN
4/1/1	100/----	B-Routed	dhcp	--	--	1	0	--

Pppoe User Id: --
1 services displayed

5 View the LAN interfaces.

```
zSH> cpe rg lan show 4/1/1
```

```

CPE  UNI      UNI-Vlan/Slan  Vlan/Slan  G-VLAN  IP-Address  IP Com  Dhcp Srvr
Rg-Mode  Profile  Profile
=====
4/1/1 eth  1              0,100/----  ----    192.168.1.1  100001  100001
B-Routed
Services displayed: 1

```

Activating the ONU

Activate the ONU to add it to the system. In Unified Service Provisioning, after changing the service configuration on an activated ONU, the services configuration will be updated automatically.

- 1 To activate an ONU first run the **gpononu show** command to display the ONUs currently on the OLT, and discover the available serial numbers.

The **gpononu show** command has options to select by slot and OLT. If users run the command without defining the slot/OLT the command will check for ONTs on every port of every card and depending on the number of cards, may take several minutes to complete.

```

zSH> gpononu show 4/1
Free ONUs for slot 4 olt 1:
   1   2   3   4   5   6   7   8   9  10  11  12
  13  14  15  16  17  18  19  20  21  22  23  24
  25  26  27  28  29  30  31  32  33  34  35  36
  37  38  39  40  41  42  43  44  45  46  47  48
  49  50  51  52  53  54  55  56  57  58  59  60
  61  62  63  64
Discovered serial numbers for slot 4 olt 1:
sernoID  Vendor  Serial Number      Time Discovered
   1      ZNTS   03194B28          JUL 10 01:11:18 2014

```

- 2 Run the **gpononu set** command to associate a ONU port ID to a discovered ONT's serial number:

```

zSH> gpononu set 4/1/1 1
Onu 1 successfully enabled with serial number ZNTS
03194B28

```

- 3 Run the **gpononu show** command to verify the ONT is enabled, and OMCI support is added into the ONT (the associated internal ME profile can be displayed).

```

zSH> gpononu show 4/1/1
ONU      Name      Enabled Model # Number      Serial      OMCI files and profiles
=== =====
   1      1-4-1-1  Yes      2426 ZNTS 03194B28  ME  zhone-2426
Note: NULL Model String indicates not able to get model ID

```

- 4 Run the **gpononu status** command to verify the OMCI Config State is active.

```
zSH> gpononu status 4/1/1
```

ID	Onu	OperStatus	ConfigState	Download State	OLT Rx Power	ONT Rx Power	Distance (KM)	GPON OnuStatus
1	1-4-1-1	Up	Active	NoUpgrade	-19.2 dBm	-20.0 dBm	18	Active

Performing other necessary USP Data related configuration

To perform the other Data related configuration, such as creating CPE Ethernet subscriber profile, refer to [Create High Speed Internet on Dynamic OMCI with Uplink and Downlink on page 748](#).

Creation of Video Services in RG

This section shows how to create video services on a zNID Ethernet Uni-port with rg-bridged mode.

- [Creating a GPON traffic profile, page 803](#)
- [Creating uplink/downlink MXK bridges, and CPE connections in RG-bridged mode for Video Services in RG, page 803](#)
- [Performing other necessary Video related configuration, page 805](#)

Creating a GPON traffic profile

The GPON traffic profile is required for the GPON bridge add command.

```
zSH> new gpon-traffic-profile 2
gpon-traffic-profile 2
Please provide the following: [q]uit.
    guaranteed-upstream-bw: -> {0}: 512
    traffic-class: -----> {ubr}:
compensated: -----> {false}:
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: ----> {0}:
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Creating uplink/downlink MXK bridges, and CPE connections in RG-bridged mode for Video Services in RG

In the example below, Ethernet Uni-port 2 in RG is member of the RG-bridged VLAN 200.

1 Create an uplink bridge interface on the MXK

```
zSH> bridge add 1-a-5-0/eth uplink vlan 200
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-200/bridge
bridge-path added successfully
```

2 Create a downlink MXK bridge, and a connection between the ONT 1-4-1-1 GEM port 403 and the CPE untagged UNI Eth 2 on VLAN 200.

```
zSH> bridge add 1-4-1-1/gpononu gem 403 gtp 2 downlink vlan 200 tagged video 0/4
eth 2 rg-bridged
Adding bridge on 1-4-1-1/gpononu
Created bridge-interface-record 1-4-1-403-gponport-200/bridge
CPE Connection 1-4-1-403/gponport/12/2/0/0 has been created
```

The first part of this command, “**bridge add 1-4-1-1/gpononu gem 403 gtp 2 downlink vlan 200 tagged video 0/4**” creates a new MXK bridge. The second part of this command, “**eth 2 rg-bridged**” creates a CPE connection in CPE 1-4-1-1 to bridge untagged UNI eth 2 to GEM port 403, a bridged WAN-side interface in RG, and a bridged LAN-side interface in RG.

3 View the bridges.

```
zSH> bridge show
```

Orig	Type	VLAN/SLAN	Physical	Bridge	St
dwn	Tagged	100	1/4/1/1/gpononu	1-4-1-303-gponport-100/	
bridge	UP	D 00:02:71:19:4b:28			
dwn	Tagged	200	1/4/1/1/gpononu	1-4-1-403-gponport-200/	
bridge	UP				
upl	Tagged	100	1/a/5/0/eth	ethernet5-100/bridge	UP
S VLAN 100 default					
upl	Tagged	200	1/a/5/0/eth	ethernet5-200/bridge	UP
S VLAN 200 default					

4 Bridge Interfaces displayed

4 View the WAN interfaces.

```
zSH> CPE> RG> WAN> show 4/1/1
```

CPE	VLAN/SLAN	RG Mode	IP Address	Auth	Retry Interval	Ip-Com Profile	Port-Fwd List Profile	G-VLAN
4/1/1	100/----	B-Routed	dhcp	--	--	1	0	--
Pppoe User Id: --								
4/1/1	200/----	Bridged	IP Unconfigured	--	--	0	0	--
Pppoe User Id: --								

2 services displayed

5 View the LAN interfaces.


```
zSH> cpe rg lan show 4/1/1
```

CPE	UNI	UNI-Vlan/Slan	Vlan/Slan	G-VLAN	IP-Address	IP Com Profile	Dhcp Srvr Profile
Rg-Mode							
4/1/1	eth 1		0,100/----	----	192.168.1.1	100001	100001
B-Routed							
4/1/1	eth 2		0,200/----	----	0.0.0.0	0	0
Bridged							
Services displayed: 2							

Performing other necessary Video related configuration

To perform the other video related configuration, such as CPE Video Access Control profile, CPE Video profile, CPE Ethernet subscriber profile, refer to [Create Uplink and Downlink Bridges on Dynamic OMCI for Video on page 755](#).

Creation of Voice Services in RG

This section shows how to create voice services on all the POTS ports in a zNID with rg-bridged mode.

- [Creating a GPON traffic profile, page 805](#)
- [Creating uplink/downlink MXK bridges, and CPE connections in RG-bridged mode for SIP services in RG, page 806](#)
- [Creating IP Common profile for Voice services \(Optional\), page 807](#)
- [Performing other necessary Voice related configuration, page 807](#)
- [Creating SIP-PLAR services in RG, page 808](#)
- [Creating MGCP services in RG, page 809](#)

Creating a GPON traffic profile

The GPON traffic profile is required for the GPON bridge add command.

```
zSH> new gpon-traffic-profile 3
```

```
gpon-traffic-profile 3
Please provide the following: [q]uit.
  guaranteed-upstream-bw: -> {0}: 512
  traffic-class: -----> {ubr}:
compensated: -----> {false}:
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: ----> {0}:
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
```

```
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved..
```

Creating uplink/downlink MXK bridges, and CPE connections in RG-bridged mode for SIP services in RG

In the example below, the SIP service is created, and POTS ports in RG are members of the RG-bridged VLAN 300.

- 1 Create an uplink bridge interface on the MXK

```
zSH> bridge add 1-a-5-0/eth uplink vlan 300
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-300/bridge
bridge-path added successfully
```

- 2 Create a downlink MXK bridge, and a connection between the ONT 1-4-1-1 GEM port 501 and the CPE UNI POTS ports on VLAN 300.

```
zSH> bridge add 1-4-1-1/gpononu gem 501 gtp 3 downlink vlan 300 tagged rg-bridged sip
Adding bridge on 1-4-1-1/gpononu
Created bridge-interface-record 1-4-1-501-gponport-300/bridge
CPE Connection 1-4-1-501/gponport/14/0/0/0 has been created
```

The first part of this command, “**bridge add 1-4-1-1/gpononu gem 501 gtp 3 downlink vlan 300 tagged**” creates a new MXK bridge. The second part of this command, “**rg-bridged sip**” creates a CPE connection in CPE 1-4-1-1 to bridge the UNI POTS ports to GEM port 501, and a bridged WAN-side interface in RG to be used for the internal voice client. Note that system default interface and system DNS client are automatically set to the Voice VLAN.

- 3 View the bridges.

```
zSH> bridge show
      Orig
Type VLAN/SLAN  VLAN/SLAN      Physical      Bridge
St  Table Data
-----
dwn      Tagged  100      1/4/1/1/gpononu      1-4-1-303-gponport-100/
bridge      UP  D 00:02:71:19:4b:28
dwn      Tagged  200      1/4/1/1/gpononu      1-4-1-403-gponport-200/
bridge      UP
dwn      Tagged  300      1/4/1/1/gpononu      1-4-1-501-gponport-300/
bridge      UP  D 00:02:71:19:4b:29

D 00:02:71:19:4b:28
upl      Tagged  100      1/a/5/00,100/----/eth      ethernet5-100/
bridge      UP  S VLAN 100 default
upl      Tagged  200      1/a/5/0/eth      ethernet5-200/bridge
UP  S VLAN 200 default
```

```

upl          Tagged 300 1/a/5/0/eth          ethernet5-300/bridge
UP S VLAN 300 default
6 Bridge Interfaces displayed

```

4 View the WAN interfaces.

```
zSH> CPE> RG> WAN> show 4/1/1
```

CPE	VLAN/SLAN	RG Mode	IP Address	Auth	Retry Interval	Ip-Com Profile	Port-Fwd List Profile
G-VLAN							
4/1/1	100/----	B-Routed	dhcp	--	--	1	0
Pppoe User Id: --							
4/1/1	200/----	Bridged	IP Unconfigured	--	--	0	0
Pppoe User Id: --							
4/1/1	300/----	Bridged	dhcp	--	--	1	0
Pppoe User Id: --							

3 services displayed

5 View the LAN interfaces.

```
zSH> CPE> RG> LAN> show 4/1/1
```

CPE	UNI	UNI-Vlan/Slan	Vlan/Slan	G-VLAN	IP-Address	IP Com Profile	Dhcp Srwr Profile
Rg-Mode							
4/1/1	eth 1		0,100/----	----	192.168.1.1	100001	100001
B-Routed							
4/1/1	eth 2		0,200/----	----	0.0.0.0	0	0
Bridged							
4/1/1	pots		0,300/----	---	---	---	Bridged

Services displayed: 3

Creating IP Common profile for Voice services (Optional)

This step is optional. If zNID is using DHCP (default setting) to get voice host IP address, users can skip this step.

If users want to assign static IP address to the zNID, users need to create a static IP common profile on WAN interface and assign the static IP address to the WAN interface. For the detail configuration, see [Static configuration on the WAN side interfaces \(without DHCP\)](#), page 834.

Performing other necessary Voice related configuration

To perform the other voice related configuration, such as CPE VoIP server profile, CPE VoIP feature profile, CPE VoIP media profile, CPE SIP Dial Plans, CPE VoIP subscriber profile, refer to [Create VoIP on Dynamic OMCI on Uplink and Downlink Bridges](#) on page 763.

Creating SIP-PLAR services in RG

- 1 Add a "sipplar" bridge in rg-bridged mode.

```
zSH> bridge add 1-9-1-16/gpononu gem 516 gtp 1 vlan 200 downlink tagged sipplar
rg-bridged
```

- 2 Create a cpe-voip-server with a valid primary-server and the signalling-protocol as "sipplar". This example created a cpe-voip-server "3".

```
zSH> CPE> VOIP> SERVER> add sipplar-server-test primary-server 10.235.9.2
signalling-protocol sipplar
```

```
zSH> cpe voip server show 3
```

Cas Events		Signalling Oob Dtmf Oob Cas Dtmf Events	
Index	Profile Name	Protocol	Events
3	sipplar-server-test	sipplar	disabled disabled
rfc4733			rfc4733

```
Primary Server      : 10.235.9.2
Secondary Server    : 0.0.0.0
Sip Domain          :
Sip Registrar       :
Mgc Termination Id Base :
Softswitch          :
OutBound Server     :
Port Id             : -1
Rtp Dscp             : 0
Signalling Dscp      : 0
Sip Reg Exp Time     : 3600
Rereg Head Start Time : 360
Sip Reg Retry Time   : 60
Release Timer        : 10
Roh Timer            : 15
1 entries found.
```

- 3 Note that SIP PLAR CPE VoIP connection requires dial numbers and usernames.

This example add SIP PLAR services on ONU 9/1/16 Ethernet port 1, and specified dial-number, user-name and VoIP server profile as 3.

```
zSH> CPE> VOIP> add 9/1/16/1 dial-number 7311002 username 7311002
voip-server-profile 3
```

```
zSH> CPE> VOIP> add 9/1/16/2 dial-number 7311003 username 7311003
voip-server-profile 3
```

- 4 Create a ip-com profile, if it is a static configuration create a valid gateway.

If DHCP is used, Step 4 & 5 are not needed.

```
zSH> CPE> RG> WAN> IP-COM> add sipplar-ip host-ip-option static gateway
172.24.200.52 netmask 255.255.255.0
```

```
zSH> cpe rg wan ip-com show all
```

secure	default			host IP			
Index	Profile Name			option	gateway	igmp fn	nat
fwd	iface	dns src	dns ty				
pe							
=====	=====	=====	=====	=====	=====	=====	=====
=====	=====	=====	=====				
==							
4	sipplar-ip			static	172.24.200.52	none	nat
disabl	false	false	prox				
y							

- 5 By default in the shell CPE>RG>WAN> ip address will be dhcp. If users have a static set up users need to modify it.

```
zSH> cpe rg wan show 9/1/16
```

CPE	VLAN/SLAN	RG Mode	IP Address	Auth	Retry Interval	Ip-Com Profile	Port-Fwd List	Profile
G-VLAN								
=====	=====	=====	=====	=====	=====	=====	=====	=====
=====								
9/1/14	200/----	Bridged	dhcp	--	--	0		0
--								
Pppoe User Id: --								
9/1/14	250/----	B-Routed	dhcp	--	--	1		0
--								
Pppoe User Id: --								
2 services displayed								

```
zSH> cpe rg wan modify 9/1/16 vlan 200 ip-addr 172.24.200.51 ip-com-profile 4
Service has been modified
```

Creating MGCP services in RG

After configured IP interface, VoIP signalling protocol, and VoIP server settings property, user can create POTS to MGCP softswitch configuration in RG.

The following example creates a POTS to MGCP software connection.

- 1 Verify/Modify the **countryregion** parameter of the **system 0** profile ensures that the country-specific voice settings are correctly set. (Its one time setup)
- 2 Use the bridge add command with mgcp and rg-bridged keywords to create a path to MGCP in rg-bridged mode.

```
zSH> bridge add 1-1-1-3/gpononu gem 401 gtp 1 vlan 500 downlink tagged rg-bridged
mgcp
Adding bridge on 1-1-1-3/gpononu
Created bridge-interface-record 1-1-1-401-gponport-500/bridge
CPE Connection 1-1-1-401/gponport/22/0/0/0 has been created
```

```
zSH> bridge show 1/1/1
      Orig
Type   VLAN/SLAN  VLAN/SLAN  Physical  Bridge
St  Table Data
-----
dwn                Tagged  105    1/1/1/1/gpononu  1-1-1-301-gponport-105/
bridge            DWN
dwn                Tagged  500    1/1/1/3/gpononu  1-1-1-401-gponport-500/
bridge            DWN
2 Bridge Interfaces displayed
```

```
zSH> bridge show onu 1-1-1-3
      GEM  ONU  DSCP  ONU UNI  OLT  OLT
ONU      Port UNI  to COS VLAN/SLAN VLAN/SLAN  G-VLAN MVR  Service Rg-Mode
OLT Bridge                               ST
-----
1/1/3    401  pots  ----- /----- Tagged  500  ----- mgcp  Bridged
1-1-1-401-gponport-500/bridge                               DWN
1 Bridge Interfaces displayed
1 GPON ONU Connections displayed
```

- 3 Create the MGCP server.

MGCP signaling establishes call control elements or call agents to handle call control. MGCP devices execute the commands sent by the call agents.

Use the **cpe voip server add** command with **mgcp** keyword to create a MGCP server. Here are the MGCP related parameters in the VoIP server settings. For the detail explanation of each parameter, refer to [Table 73](#) in the *MXK Configuration Guide*.

- primary-server: Specify the MGCP server (i.e. call agent) IP address
- mgcp-client-address-mode: IP (default value), IPbracketed, or domainname. IP and IPbracketed will cause the MGCP client name to be the bound voice host IP address. Domainname will allow the users to input any text string for accessing the call agent, usually a domain name. Most customers use IP or IPbracketed.

- signalling-dscp
- mgcp-persistent-notify
- a** This example creates the mgcp-test MGCP server with IP address 172.15.80.15, and it uses the default value IP as mgcp-client-address-mode.

```
zSH> CPE> VOIP> SERVER> add mgcp-test primary-server 172.15.80.15
signalling-protocol mgcp
```

Profile "mgcp-test" has been created with index 2

```
zSH> CPE> VOIP> SERVER> show mgcp-test
```

```

                                     Signalling Oob Dtmf Oob Cas  Dtmf Events
Cas Events
  Index          Profile Name          Protocol  Events  Events  Passing
Method Passing Method
=====
=====
      2              mgcp-test      mgcp disabled disabled      rfc4733
rfc4733
Primary Server      : 172.15.80.15
Secondary Server    : 0.0.0.0
Sip Domain          :
Sip Registrar       :
Mgc Termination Id Base :
Softswitch          :
OutBound Server     :
Port Id             : -1
Rtp Dscp             : 0
Signalling Dscp     : 0
Sip Reg Exp Time    : 3600
Rereg Head Start Time : 360
Sip Reg Retry Time  : 60
Release Timer       : 10
Roh Timer           : 15
MGCP Address Mode   : ip
MGCP Persistent Notify : disabled
1 entries found.
```

- b** If users want to use domain name as the mgcp-client-address-mode, users can use this step, otherwise skip it.

Create a MGCP server with domainname as the mgcp-client-address-mode, and then specify a domain name in the CPE system.

```
zSH> CPE> VOIP> SERVER> add mgcp-test primary-server 172.15.80.15
signalling-protocol mgcp mgcp-client-address-mode domainname
```

Profile "mgcp-test" has been created with index 2

```
zSH> CPE> SYSTEM> add 1/1/3 mgcp-client-name 1234.zhone.com
```

System profile "1/1/3" has been created

```
zSH> CPE> SYSTEM> show 1/1/3
```

```
CPE          SYSTEM COMMON PROFILE          MGCP CLIENT NAME
=====
1/1/3  Default_Cpe_System_Common/1  1234.zhone.com
1 entries found.
```

4 Use the **CPE VOIP media add** command to configure the VoIP service settings.

The following parameters in this command can be used for MGCP services. For the detail explanation, refer to [Table 76](#) in the *MXK Configuration Guide*.

- packet-period-selection-first order: Voice sample size in ms. Specify the time that the DSP will encode voice before sending. The longer the time the more propagation delay in the data stream, but also the more efficient the packetization.
- silence-suppression-first order: Enable or disable Silence Suppression.
- echo-cancel: Enable or disable Echo Cancellation
- fax-mode: Specify the fax mode is pass-through or t38.
- codec-selection-n-order: Specifies the codec selection as defined by RFC 3551. n is in the range of “first-order” to “fourth-order”.

```
zSH> CPE> VOIP> MEDIA> add MediaConfig fax-mode t38 codec-selection-first-order
pcmu codec-selection-second-order pcma codec-selection-third-order g729
codec-selection-fourth-order g722
Profile "MediaConfig" has been created with index 2
```

```
zSH> CPE> VOIP> MEDIA> show all
```

Packet-period Index selection	silence Profile Name suppression	echo cancel	fax mode	codec Selection
1 (1st)	Default_Cpe_Voip_Media 10 (1st) disabled (1st)	enabled	passThrough	PCMU
(2nd)	10 (2nd) disabled (2nd)			PCMU
(3rd)	10 (3rd) disabled (3rd)			PCMU
(4th)	10 (4th) disabled (4th)			PCMU
2 (1st)	MediaConfig 10 (1st) disabled (1st)	enabled	T38	PCMU
(2nd)	10 (2nd) disabled (2nd)			PCMA
(3rd)	10 (3rd) disabled (3rd)			G729
(4th)	10 (4th) disabled (4th)			G722

2 entries found.

- 5** Use the **CPE VOIP add** command to add the POTS to MGCP connection.

Here are the MGCP related parameters in this command. For the detail explanation, refer to [Table 77](#) in the *MXK Configuration Guide*.

- admin-state: Up or Down
- dial-number: A text field that specifies subscriber directory number, up to 36 byte character string.
- user-name: A text field that identifies the port to the switch. Up to 25 byte unique character string. Note that username field is required for MGCP configuration.
- tx-gain: A gain value for the transmit signal.
- rx-gain: A gain value for the received signal.
- phone-follows-wan: When enabled the phone will lose power any time the WAN is operation status of down. This will allow line monitoring equipment to detect loss of services.

The following examples create MGCP services on ONU 1/1/3 POTS1 and POTS 2, specify the user names, and associate VoIP server profile mgcp-test and VoIP media profile MGCP Feature with them.

```
zSH> CPE> VOIP> add 1/1/3/1 username 1111 voip-server-profile mgcp-test
voip-media-profile MediaConfig
Service has been created
```

```
zSH> CPE> VOIP> add 1/1/3/2 username 1112 voip-server-profile mgcp-test
voip-media-profile MediaConfig
Service has been created
```

- 6** By default, DHCP is used. This must be changed since static assignment of the IP address is nearly always used. To create a valid gateway for static configuration with the WAN ip-common profile, use the [Step 6](#) and [Step 7](#).

```
zSH> CPE> RG> WAN> IP-COM> add mgcp-ip host-ip-option static gateway 172.24.200.52
netmask 255.255.255.0
Profile "mgcp-ip" has been created with index 3
```

```
zSH> CPE> RG> WAN> IP-COM> show mgcp-ip
```

IP	Index	Profile Name	secure default	option	gateway	igmp fn	nat
mgcp-ip	3	mgcp-ip	static	172.24.200.52	none	nat	

```

disabl  false  false  default
Net Mask: 255.255.255.0
Primary Dns: 0.0.0.0
Secondary Dns: 0.0.0.0
Firewall Access: ping
```

1 entries found.

- 7 By default RG WAN IP address will use DHCP and associated with the default IP-Common profile 1. This must be changed to assign the Static IP Address to be used by the MGCP voice client.

Creation of Data services on Wireless interfaces

This section shows how to create data services on a zNID Wireless interface with rg-brouted mode.

- [Creating uplink/downlink MXK bridges, and CPE connections in RG-brouted mode, page 814](#)
- [Creating CPE WLAN subscriber profile and associate it with a CPE WLAN common profile or CPE WLAN common advance profile \(optional\), page 815](#)
- [Creating CPE WLAN common profile \(optional\), page 818](#)
- [Creating CPE WLAN common advance profile \(optional\), page 821](#)

Creating uplink/downlink MXK bridges, and CPE connections in RG-brouted mode

This example creates the data service on WLAN 1 interface (i.e. SSID 0 in the ONT WebUI). This WLAN interface is using same VLAN and GEM port as Uni Ethernet 1’s.

- 1 This example is not going to create uplink interface. It uses the VLAN that was already created on the uplink for data services.
- 2 Create a downlink MXK bridge, and a connection between the ONT 1-4-1-1 GEM port 303 and the CPE LAN side wireless interface 1 (i.e. SSID 0 in the ONT WebUI) on VLAN 100 for untagged packets.

```
zSH> bridge add 1-4-1-1/gpononu gem 303 gtp 1 downlink vlan 100 tagged wlan 1
rg-brouted
CPE Connection 1-4-1-303/gponport/18/1/0/0 has been created
```

The first part of this command “**bridge add 1-4-1-1/gpononu gem 303 gtp 1 downlink vlan 100 tagged**” indicates the MXK bridge. The second part of this command “**wlan 1 rg-brouted**” creates another CPE UNI connection with this MXK bridge, and a routed LAN side wireless interface in RG. A CPE-WLAN-Subscriber profile will be created too.

- 3 View the bridges.

```
zSH> bridge show
```

Orig							
Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table	Data
dwn		Tagged 100	1/4/1/1/gpononu	1-4-1-303-gponport-100/bridge	UP	D	00:02:71:19:4b:28
dwn		Tagged 200	1/4/1/1/gpononu	1-4-1-403-gponport-200/bridge	UP		
dwn		Tagged 300	1/4/1/1/gpononu	1-4-1-501-gponport-300/bridge	UP	D	00:02:71:19:4b:29
upl		Tagged 100	1/a/5/0/eth	ethernet5-100/bridge	UP	S	VLAN 100 default
upl		Tagged 200	1/a/5/0/eth	ethernet5-200/bridge	UP	S	VLAN 200 default

```
upl          Tagged 300 1/a/5/0/eth          ethernet5-300/bridge          UP S VLAN 300 default
6 Bridge Interfaces displayed
```

```
zSH> bridge show onu
```

ONU ST	GEM Port	ONU UNI	DSCP to COS	ONU VLAN/SLAN	UNI VLAN/SLAN	OLT G-VLAN	OLT MVR	Service	Rg-Mode	OLT Bridge
4/1/1 DWN	403	eth 2		Tagged	200	----		iptv	Bridged	1-4-1-403-gponport-200/bridge
4/1/1 UP	303	eth 1		Tagged	100	----		data	B-Routed	1-4-1-303-gponport-100/bridge
4/1/1	303	wlan 1		Tagged	100	----		data	B-Routed	1-4-1-303-gponport-100/bridge
4/1/1 UP	501	pots		Tagged	300	----		sip	Bridged	1-4-1-501-gponport-300/bridge

```
3 Bridge Interfaces displayed
4 GPON ONU Connections displayed
```

4 View the WAN interfaces.

```
zSH> CPE> RG> WAN> show 4/1/1
```

CPE	VLAN/SLAN	RG Mode	IP Address	Auth	Retry Interval	Ip-Com Profile	Port-Fwd List Profile	G-VLAN
4/1/1	100/----	B-Routed	dhcp	--	--	1	0	--
Pppoe User Id: --								
4/1/1	200/----	Bridged	IP Unconfigured	--	--	0	0	--
Pppoe User Id: --								
4/1/1	300/----	Bridged	dhcp	--	--	1	0	--
Pppoe User Id: --								

```
3 services displayed
```

5 View the WLAN interfaces.

```
zSH> CPE> RG> LAN> show 4/1/1
```

CPE	UNI	UNI-Vlan/Slan	Vlan/Slan	G-VLAN	IP-Address	IP Com Profile	Dhcp Srvr Profile	Rg-Mode
4/1/1	eth 1		Tagged 100	----	192.168.1.1	100001	100001	B-Routed
4/1/1	eth 2		Tagged 200	----	0.0.0.0	0	0	Bridged
4/1/1	pots		Tagged 300	----	0.0.0.0	0	0	Bridged
4/1/1	wlan 1		Tagged 100	----	192.168.1.1	100001	100001	B-Routed

```
Services displayed: 4
```

Creating CPE WLAN subscriber profile and associate it with a CPE WLAN common profile or CPE WLAN common advance profile (optional)

By default, the admin-state of the CPE Wireless LAN (WLAN) UNI port is up after creation of CPE connection on that CPE WLAN UNI port. Because of that, the Data and Video traffic can run on this WLAN UNI port without further configuration.

If users want to change the default WLAN physical configurations, users can use the **cpe wlan add** command. With this command, the CPE WLAN subscriber profile is created *manually*. The WLAN UNI port ID specified in

this command must match the one assigned in the **bridge add** command when creating downlink bridge and CPE connection.

Note that a CPE WLAN subscriber profile will also be created *automatically* if users set the ONU WLAN port's admin-state with the **gpononu port** command. For the details on those command, refer to [Administration of subscriber facing ports on page 914](#).

After a CPE WLAN subscriber profile is created, if users want to change the settings in that profile, users can use the **cpe wlan modify** command, which has the same command syntax as the **cpe wlan add** command.

Command:

```
cpe wlan add <interface>/<port number>
    [ admin-state < up | down > ]
    [ ssid < value> ]
    [ encrypt-key < value> ]
    [ device-pin < value > ]
    [ radius-key < value > ]
    [ wlan-com-profile < index | profile-name > ]
    [ wlan-com-adv-profile < index | profile-name > ]
```

Create a WLAN service. <interface> and <port number> must be provided.

[Table 82](#) provides the description for command options in the **cpe wlan add** command.

Table 82: cpe wlan add Command Option Explanations

Command Option	Description
<i>interface/port number</i>	ONU port ID and WLAN UNI port ID of the physical interfaces.
admin-state <i>value</i>	Activates or deactivates the functions performed by the wireless port for this subscriber. Possible values are up, down. Default value is up.
ssid <i>value</i>	Assigns the Service Set Identifier (SSID) to the wireless LAN interface. An SSID is the public name of a wireless local area network. All wireless devices on a wireless local area network must employ the same SSID in order to communicate with each other. It could be 32 characters string or less.
encrypt-key <i>value</i>	Sets the wireless encryption key on the wireless network to increase the security. The two standard types of wireless keys support Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA) encryption: • If it is a WEP 64-bit encryption key: the value could be 5 ASCII characters or 10 hexadecimal digits • If it is a WEP 128-bit encryption key: the value could be 13 ASCII characters or 26 hexadecimal digits • If it is a WPA Passphrase: the value could be 64 characters

Table 82: cpe wlan add Command Option Explanations

Command Option	Description
device-pin <i>value</i>	Sets the device pin only when Wi-Fi Protected Setup (WPS) security method is enabled. And this option is only for WLAN UNI port 1.
radius-key <i>value</i>	Sets the Remote Authentication Dial In User Server (RADIUS) authentication key. This field cannot contain a SPACE and is returned as a string of asterisks.
wlan-com-profile <i>index</i> <i>profile-name</i>	Associated CPE WLAN common profile with this WLAN UNI port. Default: 1. It indicates the default WLAN common profile is used.
wlan-com-adv-profile <i>index</i> <i>profile-name</i>	Associated CPE WLAN common advance profile with this WLAN UNI port. Default: 1. It indicates the default WLAN common advance profile is used.

To create a CPE WLAN subscriber profile with the **cpe wlan add** command:

- 1 This example sets the SSID and encrypt-key of the WLAN UNI port 1 on the ONU 4/1/1.

Note that this example enters CPE command shell; zSH> CPE> WLAN>.

```
zSH> CPE> WLAN> add 4/1/1/1 ssid zdev encrypt-key 1234567890
Service has been created
```

- 2 Show the settings of the CPE WLAN subscriber Profile for the WLAN UNI port 1 on the ONU 4/1/1. As shown in the example, a default WLAN common profile and a default WLAN Common Advanced profile with index 1 are assigned to this WLAN subscriber profile.

```
zSH> CPE> WLAN> show 4/1/1/1
```

CPE	WLAN	Admin	SSID	WLAN Com Prof	WLAN Com Adv Prof
4/1/1	1	up zdev		1	1

1 services displayed

- 3 Show all the services created on the ONU.

```
zSH> cpe show 4/1/1
CPE 4/1/1
Service: DATA
```

GEM	UNI	UNI-VLAN/SLAN	VLAN/SLAN	G-VLAN	Admin	Oper	Rg-Mode
303	eth 1		Tagged 100	0	up	up	B-Routed
303	wlan 1		Tagged 100	0	up	up	B-Routed

```
Service: IPTV
```

GEM	UNI	UNI	VLAN/SLAN	OLT	VLAN/SLAN	G-VLAN	Admin	Oper	Video Prof	Rg-Mode
403	eth 2				Tagged 200	0	up	down		Bridged

```
Service: SIP
```

GEM	UNI	UNI	VLAN/SLAN	OLT	VLAN/SLAN	G-VLAN	Host IP	IP Srvr Prof
501	pots				Tagged 300	0		1

Creating CPE WLAN common profile (optional)

This step is optional. Creation of Data bridge in step [Creating uplink/downlink MXK bridges, and CPE connections in RG-brouted mode on page 814](#) is sufficient for creating a data service. Users can use the CPE WLAN common profile if users want to add some common data service attributes.

The CPE WLAN common profile in this section covers the common settings would be used for all WLAN ports. The CPE WLAN common advance profile in the next section covers the advanced settings that could be used only for WLAN port 1.



Note: CPE WLAN common profile can only be deleted when it is not associated with any CPE WLAN subscriber profiles.

Command:

```
cpe wlan common add <profile-name <string>>
    [ net-authen < open | shared | 802dot1x | wpa |
wpa-psk | wpa2 | wpa2-psk | mixed-wpa2-wpa |
mixed-wpa2-wpa-psk > ]
    [ hide-ap < enabled | disabled > ]
    [ isolate-clients < enabled | disabled > ]
    [ wmm-advertise < enabled | disabled > ]
    [ mcast-forward < enabled | disabled > ]
    [ max-clients < value > ]
    [ wpa-group-rekey-interval < value > ]
    [ wpa-encryption < aes | tkip-aes > ]
    [ wep-encryption < enabled | disabled > ]
    [ wep-strength < 64bits | 128bits > ]
    [ radius-server-ip < IP address > ]
    [ radius-port < value > ]
    [ wpa2-preauth < enabled | disabled > ]
    [ reauthen-interval < value > ]
```

This command creates a new profile. The <profile-name> must be supplied and must be unique for the profile type. The profile index will be automatically generated.

[Table 83](#) provides the description for command options in the **cpe wlan common add** command.

Table 83: cpe wlan common add Command Option Explanations

Command Option	Description
profile-name <string>	Specifies a unique CPE WLAN common profile name. 36 characters string.
net-authen value	Configure the network authentication method. Values: open shared 802dot1x wpa wpa-psk wpa2 wpa2-psk mixed-wpa2-wpa mixed-wpa2-wpa-psk
hide-ap value	Enable or disable the suppression of the advertising of the access point's SSID. If enabled, clients will need to configure the SSID to associate. Values: enabled disabled Default: disabled
isolate-clients value	Isolate clients within the wireless network from communicating directly with each other. Values: enabled disabled Default: disabled
wmm-advertise value	Wireless Multi Media (WMM) provides a subset of the IEEE 802.11e QoS standard, which adds prioritization to wireless to optimize their performance. When multiple concurrent applications are on the wireless network each application may have different latency and throughput needs. WMM provides for this optimization, however WMM may provide slower. Values: enabled disabled Default: disabled
mcast-fwd value	Wireless Multicast Forwarding enables the ability to send wireless packets to be intercepted by all nodes in the transmission range of the sender. Values: enabled disabled Default: disabled

Table 83: cpe wlan common add Command Option Explanations

Command Option	Description
max-clients <i>value</i>	The maximum number of wireless client devices that may be simultaneously connected to the wireless network. Values: 1-50 Default: 16
wpa-group-rekey-interval <i>value</i>	Specifies WPA Group Rekey Interval. Values: 0-9999999999 Default: 0
wpa-encryption <i>value</i>	WPA encryption mode. Values: aes tkip-aes Default: aes
wep-encryption <i>value</i>	WEP encryption mode. Values: enabled disabled Default: disabled
wep-strength <i>value</i>	WEP encryption strength. Values: bit64 bit128 Default: bit128
radius-server-ip <i>value</i>	IP address of the Remote Authentication Dial In User Server (RADIUS) used for 802.1x authentication. Default: 0.0.0.0
radius-port <i>value</i>	UDP port to use for accessing the Remote Authentication Dial In User Server (RADIUS). Values: 0-9999999999 Default: 1812
wpa2-preauth <i>value</i>	Enable or disable WPA2 pre-authentication. Values: enabled disabled Default: disabled

Table 83: cpe wlan common add Command Option Explanations

Command Option	Description
reauthen-interval <i>value</i>	WPA2 network re-authentication time, in seconds. Values: 0-999999999 Default: 36000

1 Create the CPE WLAN common profile.

```
zSH> CPE> WLAN> COMMON> add wifi net-authen wpa2-psk
Profile "wifi" has been created with index 2
```

2 Show all the CPE WLAN common profiles.

```
zSH> CPE> WLAN> COMMON> show all
```

Index	Profile Name	Net Authentication	WPA Encrypt	WPA Encrypt	WEP Strength
1	default-wlan1	open	aes	disabled	128bits
2	wifi	wpa2-psk	aes	disabled	128bits

2 entries found.

3 If users want to delete a cpe WLAN common profile, use the **cpe wlan common delete** <profile-index> | <profile-name> command.

```
zSH> CPE> WLAN> COMMON> delete 2
Profile has been deleted.
```

4 Users can use the find command to find the associated CPE WLAN subscriber profile.

This example assumes CPE WLAN common profile 1 is being associated with a CPE ethernet subscriber profile on ONU 4/1/1:

```
zSH> CPE> WLAN> COMMON> find 1
cpe-wlan-subscriber 1-4-1-1/gpononu
1 profiles displayed
```

Creating CPE WLAN common advance profile (optional)

This step is optional. Creation of Data bridge in step [Creating uplink/downlink MXK bridges, and CPE connections in RG-brouted mode on page 814](#) is sufficient for creating a data service.

The CPE WLAN common profile in the previous section covers the common settings would be used for WLAN ports 1 to 4. The CPE WLAN common advance profile in this section covers the advanced settings that could be used only for WLAN port 1.



Note: CPE WLAN common advance profile can only be deleted when it is not associated with any CPE WLAN subscriber profiles.

Command:

```
cpe wlan common advance add <profile-name <string>>
    [ channel < auto | c1 | c2 | c3 | c4 | c5 | c6 |
c7 | c8 | c9 | c10 | c11 | c12 | c13 > ]
    [ auto-chan-timer < value > ]
    [ 802dot11n-mode < auto | disabled > ]
    [ 802dot11n-rate < auto | use54g | 6.5m | 13m |
19.5m | 26m | 39m | 58.5m | 65m | 78m | 104m | 117m
| 130m > ]
    [ 802dot11n-protect < auto | disabled > ]
    [ 802dot11n-client-only < enabled | disabled > ]
    [ 54g-rate < auto | 1m | 2m | 5.5m | 6m | 9m |
11m | 12m | 18m | 24m | 36m | 48m | 54m > ]
    [ mcast-rate < auto | 1m | 2m | 5.5m | 6m | 9m |
11m | 12m | 18m | 24m | 36m | 48m | 54m > ]
    [ basic-rate < default | all | 1n2m | std-rates
> ]
    [ fragment-threshold < 256 - 2346> ]
    [ rts-threshold < 0-2347> ]
    [ dtim-interval < 1-255> ]
    [ beacon-interval < 1-65535 > ]
    [ global-max-clients < 1-128 > ]
    [ xpress-tech < enabled | disabled > ]
    [ tx-power < 1-100 > ]
    [ wmm < enabled | disabled > ]
    [ wmm-no-ack < enabled | disabled > ]
    [ wmm-apsd < enabled | disabled > ]
    [ ap-mode < accesspoint | wirelessbridge > ]
    [ bridge-restrict < enabled | enabledscan |
disabled > ]
    [ wps < enabled | disabled > ]
    [ wps-add-client-method < push-button |
station-pin| ap-pin > ]
    [ wps-ap-mode < configured | unconfigured > ]
```

This command creates a new profile. The <profile-name> must be supplied and must be unique for the profile type. The profile index will be automatically generated.

Table 83 provides the description for command options in the **cpe wlan common add** command.

Table 84: cpe wlan common advance add Command Option Explanations

Command Option	Description
profile-name <string>	Specifies a unique CPE WLAN common advance profile name. 36 characters string.
channel <i>value</i>	Defines which channel to use, or 'auto' for automatic selection of a channel with low interference. 802.11b and 802.11g use channels to limit interference from other devices. Values: auto, c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13 Default: auto
auto-chan-timer <i>value</i>	When configured for auto mode, this timer value specifies how often (in minutes) to re-analyze the spectrum to select a low interference channel. Note: auto channel rescan will only occur when there are no actively connected devices. Values: 0-2147483647 Default: 15
802dot11n-mode <i>value</i>	802.11n MIMO EWC modes of operation. 802.11n improves data rates via MIMO (multiple-input, multiple-output) using spatial streams which each have a channel width of 40 MHz or 20 MHz. Usage of 802.11n in the 2.4 and 5GHz modes should depend on interference with other 802.11 or bluetooth systems on the same frequency. Enhanced Wireless Consortium (EWC) provides extra enhancements (adding the ability to define 20 MHz channels). Values: auto disabled Default: auto
802dot11n-rate <i>value</i>	Supported 802.11n MIMO rates, in Mbps. Values: auto, use54g, 6.5m, 13m, 19.5m, 26m, 39m, 58.5m, 65m, 78m, 104m, 117m, 130m Default: auto
802dot11n-protect <i>value</i>	802.11n MIMO protection modes. Values: auto disabled Default: auto

Table 84: cpe wlan common advance add Command Option Explanations

Command Option	Description
802dot11n-client-only <i>value</i>	Enable or disable the restriction of access to 802.11n clients only. When enabled, prevent 802.11b/g clients from connecting. Values: enabled disabled Default: disabled
54g-rate <i>value</i>	The rate when the radio is operating in 802.11g mode. This parameter only applies when the Multiple Input Multiple Output (MIMO) 802.11n Rate is set to use54g. Values: auto, 1m, 2m, 5.5m, 6m, 9m, 11m, 12m, 18m, 24m, 36m, 48m, 54m Default: 1m
mcast-rate <i>value</i>	The rate for multicast traffic. Values: auto, 1m, 2m, 5.5m, 6m, 9m, 11m, 12m, 18m, 24m, 36m, 48m, 54m Default: auto
basic-rate <i>value</i>	The rate when the radio is operating in basic 802.11b/g mode. Values: default, all, 1n2m, std-rates Default: default
fragment-threshold <i>value</i>	The threshold at which packets are fragmented. Values: 256-2346 Default: 2346
rts-threshold <i>value</i>	The packet size of a request-to-send (RTS) transmission. A low threshold implies RTS packets are sent more frequently, thus requiring more bandwidth but ensuring packet transmission on a busy network. Values: 0-2347 Default: 2347
dtim-interval <i>value</i>	The interval at which Delivery Traffic Indication Messages (DTIM) are generated. A DTIM message notifies a wireless client that a packet is waiting for transmission. Values: 1-255 Default: 1
beacon-interval <i>value</i>	The interval at which Beacons are generated. Values: 1-65535 Default: 100

Table 84: cpe wlan common advance add Command Option Explanations

Command Option	Description
global-max-clients <i>value</i>	The maximum number of wireless client devices that may be simultaneously connected to the radio. This value should include the sum total of all active SSIDs. Values: 1-128 Default: 16
xpress-tech <i>value</i>	Enable or disable the XPress(TM) Technology. Values: enabled disabled Default: disabled
tx-power <i>value</i>	The percentage of total power that should be used for data transmissions. Values: 0-100 Default: 100
wmm <i>value</i>	Enable or disable Wifi Multimedia. If it is enabled, audio, video and voice application data is prioritized over other network traffic. Values: enabled disabled Default: enabled
wmm-no-ack <i>value</i>	Enable or disable the suppression of acknowledgements for frames that do not require a QOS Acknowledgement. This avoids the unnecessary transmission of acknowledgements for highly time-critical data. Values: enabled disabled Default: Disabled
wmm-apsd <i>value</i>	Enable or disable the Automatic Power Save Delivery (APSD) power management method. This feature is useful for bi-directional applications, such as VoIP phones. Values: enabled disabled Default: enabled
ap-mode <i>value</i>	Wireless access point modes of operation: access point and WDS or WDS only. Values: access-point wireless-bridge Default: access-point

Table 84: cpe wlan common advance add Command Option Explanations

Command Option	Description
bridge-restrict <i>value</i>	Wireless Bridge Restriction Modes of operation. Values: enabled enabled-scan disabled Default: disabled
wps <i>value</i>	Enable or disable WiFi Protected Setup (WPS) security method. If WPS is enabled, the network authentication method, the data encryption, and network key should also be configured in order to authenticate to this wireless network. It is available for WPA-PSK, WPA2-PSK, Mixed WPA2/WPA-PSK and Open Network Authentication methods. Values: enabled disabled Default: disabled
wps-add-client-method <i>value</i>	A client can be added via three different methods: push button, station pin or access point pin. Values: push-button sta-pin ap-pin Default: push-button
wps-ap-mode <i>value</i>	If the provider is using an external registrar for security, select "Configured". The PIN for AP mode is specified by the registrar. Provide this PIN to the client. Issue "Config AP" to begin the registration process with the client. Values: configured unconfigured Default: configured

1 Create the CPE WLAN common advance profile.

```
zSH> CPE> WLAN> COM-ADV> add plan1 802dot11n-rate auto 802dot11n-client-only
disabled 54g-rate 1m mcast-rate auto basic-rate default xpress-tech disabled
tx-power 100 wmm enabled
Profile "plan1" has been created with index 4
```

2 Show all the CPE WLAN common advance profiles.

```
zSH> CPE> WLAN> COM-ADV> show all
```

Index	Profile Name	Channel	802.11n Rate	802.11n Only	54G Rate	Mcast Rate	Basic Rate	Xpress	Tx Pwr	WMM
1	Default_Cpe_WlanAdv	auto	26m	enabled	auto	12m	all	enabled	80	disabled
2	plan1	auto	auto	disabled	1m	auto	default	disabled	100	enabled

2 entries found.

- 3 If users want to delete a CPE WLAN common advance profile, use the **cpe wlan common delete** <profile-index> | <profile-name> command.

```
zSH> CPE> WLAN> COM-ADV> delete 2
Profile has been deleted.
```

- 4 Users can use the **find** command to find the associated CPE WLAN subscriber profile.

This example assumes CPE WLAN common advance profile 1 is being associated with a CPE ethernet subscriber profile on ONU 4/1/1:

```
zSH> CPE> WLAN> COM-ADV> find 1
cpe-wlan-subscriber 1-4-1-1/gpononu
1 profiles displayed
```

Create PWE on RG on TLS Bridges

Note that RG PWE is only partially supported in USP. After a PWE VLAN is created, the users still need to finish the configuration through WebUI or Post Configuration.

To create a PWE VLAN, use the **bridge add** command with **pwe** keyword. The **rg-bridged** mode is recommended for PWE. For example:

```
zSH> bridge add 1-1-1-1/gpononu gem 901 gtp 2 tls vlan 60 tagged pwe
rg-bridged
```

CPE System Level Default Settings

The CPE system level default settings on a CPE include:

- firewall settings
- sync-cookie-protection settings
- cross VLAN routing settings
- static route list profile settings
- DNS host list profile settings
- TR-069 settings, include Username, Password, ACS URL, and etc.
- management access control of CPE login accounts: Admin, Support, and User, i.e. Passwords for the three CPE accounts.

They are listed in the CPE system level common profile. There is a default CPE system common profile that applied to all CPEs in the system. Users can modify the default settings as users desired, or create a new CPE system common profile then apply to a CPE.



Note: CPE System common profile can only be deleted when it is not associated with any CPE system profiles.

To create a CPE system common profile, use the **cpe system common add** command. To apply the new CPE system common profile to a CPE, use the **cpe system add** command.

[Table 85](#) provides the description for fields in the **cpe system common add** command. The modify command has the same syntax.

Table 85: CPE system common add Command Options Explanation

Command Option	Description
<i>profile-name</i>	The name of the CPE system common profile.
firewall value	Enable or disable firewall. Enabling firewall can protect the CPE from unwanted intrusion. When firewall is enabled, incoming connections can still be selectively allowed through firewall access and port forwarding settings. Default: enabled Values: enabled disabled
sync-cookie-protection value	Protects against malicious attackers attempting to exploit TCP handshaking. Default: enabled Values: enabled disabled
cross-vlan-routing value	If "enabled" is selected, routing between VLANs is allowed. Route table lookups ignore the VLAN ID of the ingress and egress ports. If there is a match, the packet is routed out the interface specified in the Route table, regardless of which VLAN it is a member of. If "disabled" is selected, packets will be forwarded to the configured Default Route for the VLAN that they arrived on, unless there is a Route Table match within that same VLAN. Routing of packets across VLANs is prevented, providing traffic isolation. Default: disabled Values: enabled disabled
static-route-list-profile value	Address of the static-route-list profile associated with this CPE. By default, there is no static-route-list-profile created.
dns-host-list-profile name	Index of the dns-host-list profile associated with this CPE, or 0 if none. Default: 0

Table 85: CPE system common add Command Options Explanation

Command Option	Description
tr69-inform <i>value</i>	Enable or Disable the generation of Inform messages to the TR-069 ACS (Auto Configuration Server). Default: enabled Values: enabled disabled
inform-interval <i>seconds</i>	Periodic interval (in seconds) at which Inform messages will be generated. This is a TR-069 related parameter. Uint32, Default is 300
acs-url <i>value</i>	Contains the web site address of the TR-069 ACS (e.g. http://zhone.com:6050). If the URL includes a domain name, a DNS must be reachable to resolve the domain name. 256-char string.
acs-username <i>username</i>	User name required to access the TR-069 ACS. 64-char string.
acs-password <i>password</i>	User password required to access the TR-069 ACS. 64-char string.
admin-password <i>password</i>	Password for “admin” account on the CPE. Default is blank, that means it won’t overwrite the existing default value on the CPE. 16-char string. The “admin” account has unrestricted access to change and view configuration of the CPE, and to run diagnostics.
support-password <i>password</i>	Password for “support” account on the CPE. Default is blank. 16-char string. The “support” account is used to access the CPE for maintenance and to run diagnostics, however, the support login does not have full access to all configuration screens.
user-password <i>password</i>	Password for “user” account on the ONU. Default is blank. 16-char string, The “user” account can access the CPE, view a limited subset of configuration settings and statistics, as well as, update the CPE’s software.

Configuring CPE system common profile and applying to a CPE

1 Show the default CPE system common settings:

```

zSH> CPE> SYSTEM> COMMON> show 1

                                     Sync Cookie  Cross VLAN  Static Rte
Dns Host TR69      Inform
Index      Profile Name          Firewall  Protection  Routing      List
List      Inform  Interval
=====
1      Default_Cpe_System_Common  enabled   enabled     enabled      0
0  enabled      300
Acs URL      :
Acs Username :
1 entries found.

```

2 Create a new CPE system common profile:

```
zSH> CPE> SYSTEM> COMMON> add firewallldisable_System_Common
Cpe System Common profile "firewallldisable_System_Common" has been created with
index 2
```

3 Modify default setting.

```
zSH> CPE> SYSTEM> COMMON> modify firewallldisable_System_Common firewall disabled
user-password 1234
Cpe System Common profile has been modified
```

4 Apply the new common profile to CPE 1/3/1:

```
zSH> CPE> SYSTEM> add 1/3/1 sys-common-profile firewallldisable_System_Common
System profile "1/3/1" has been created
```

```
zSH> CPE> SYSTEM> show 1/3/1
CPE      SYSTEM COMMON PROFILE
=====
1/3/1 firewallldisable_System_Common/2
1 entries found.
```

Deleting CPE system common profile

If users want to delete a CPE system common profile, make sure there is no ONU is using it first.

1 To find which CPE is using the CPE system common profile:

```
zSH> CPE> SYSTEM> COMMON> find firewallldisable_System_Common
cpe-system 1-1-3-1/gpononu
1 profiles displayed
```

2 Delete CPE system and then users can delete CPE system common profile:

```
zSH> CPE> SYSTEM> delete 1/3/1
Cpe System profile has been deleted.
```

```
zSH> CPE> SYSTEM> COMMON> delete firewallldisable_System_Common
Cpe System Common profile has been deleted.
```

CPE WAN Level IP-Common Settings**Creating a CPE IP common profile for WAN**

The default CPE IP common profile(i.e. Default_Cpe_Ip_Server) specified the DHCP as the host IP option. It indicates CPE will get the host IP address automatically from the DHCP server.



Note: CPE IP common profile can only be deleted when it is not associated by any CPE IP profiles.

Command:

```
cpe rg wan ip-com add <profile-name>
    [ host-ip-option < dhcp | static > ]
    [ netmask < value > ]
    [ gateway < IP address > ]
    [ primary-dns < IP address > ]
    [ secondary-dns < IP address > ]
    [ nat < nat | napt | disabled > ]
    [ secure-fwd < enabled | disabled > ]
    [ firewall-access < http | ping | snmp | snmptrap | ssh | telnet | all |
none > ]
    [ default-iface < true | false > ]
    [ dns-src < true | false > ]
```

Table 86: cpe rg wan ip-com add Command Option Explanations

Command Option	Description
<i>profilename</i>	Specifies a unique CPE RG WAN IP common profile.
host-ip-option <dhcp static >	Selects an IP related option. DHCP or static. If DHCP is selected, it indicates CPE will get the host IP address automatically from the DHCP server. Default: DHCP
netmask <value>	Specifies the subnet mask for IP host services.
gateway <IP address>	Specifies the default gateway address used for IP host services. Default: d0.0.0.0
primary-dns <IP address>	Specifies the primary DNS IP address. If this value is 0.0.0.0, no primary SIP DNS is defined. Default: d0.0.0.0
secondary-dns <IP address>	Specifies the secondary DNS IP address. If this value is 0.0.0.0, no second SIP DNS is defined. Default: d0.0.0.0
nat <nat napt disabled>	When NAT or NAPT is selected, NAT/NAPT function is performed to translate between the public IP address and the private addresses. It is only supported on a WAN interface Default: nat
secure-fwd <enabled disabled>	When the secure forward mode is enabled, packets are not flooded to all ports. Instead, all packets are forwarded to the port that is designated as the uplink port. In this mode, users are prevented from directly communicating with each other, and broadcast frames are discarded. Default: disabled

Table 86: cpe rg wan ip-com add Command Option Explanations

Command Option	Description
firewall-access <http ping snmp snmptrap ssh telnet all none>	Lists the protocols allowed on this interface. The firewall option in the CPE system common profile must be enabled before these settings will take effect.
default-iface <true false>	When it is true, an internally generated packet (e.g., from SNMP trap, SMTP, etc.) is sent out through this interface if the destination IP address is not defined in the route table. The default value is false. Default: false
dns-src <true false>	Specifies the DNS information source. When it is true, the interface is used by the DHCP client to obtain DNS information. Default: false

The following example creates a static CPE IP common profile for voice services:

- 1 Create a CPE IP common profile with profile-name. The profile index will be generated automatically.

```
zSH> CPE> RG> WAN> IP-COM> add IPserver host-ip-option static netmask
255.255.255.0 gateway 172.168.3.254 primary-dns 172.168.19.1
Profile "IPserver" has been created with index 2
```

- 2 Show the default CPE IP common profiles and user-created CPE IP common profiles.

```
zSH> CPE> RG> WAN> IP-COM> show all
```

Index	Profile Name	host IP		gateway	igmp fn	secure default		dns src	dns type	
		option				nat	fw			iface
1	Default_Cpe_Ip_Server	dhcp		0.0.0.0	none	nat	disabl	false	false	default
2	IPserver	static		172.168.3.254	none	nat	disabl	false	false	default
100001	default-lan-ip-server100001	static		0.0.0.0	none	nat	disabl	false	false	default

3 entries found.

- 3 If users want to delete a user-created CPE IP common, use the delete command with the profile index or profile name.

```
zSH> CPE> IP> IP-COM> delete IPserver
Profile has been deleted.
```

CPE LAN Level IP-Common Settings

Creating a CPE IP common profile for LAN

The default CPE IP common profile (i.e.default-lan-ip-server100001) specified static as the host IP option. It indicates CPE will get the static host IP address.



Note: CPE IP common profile can only be deleted when it is not associated by any CPE IP profiles.

Command:

```
cpe rg lan ip-com add <profile-name>
    [ host-ip-option < dhcp | static > ]
    [ netmask < value > ]
    [ gateway < IP address > ]
    [ primary-dns < IP address > ]
    [ secondary-dns < IP address > ]
    [ igmp-function < none | snooping | proxy | snoopingproxy> ]
    [ firewall-access < http | ping | snmp | snmptrap | ssh | telnet | all |
none > ]
    [ dns-type < default | static | proxy > ]
```

Table 87: cpe rg lan ip-com add Command Option Explanations

Command Option	Description
<i>profilename</i>	Specifies a unique CPE RG LAN IP common profile.
host-ip-option < dhcp static >	Selects an IP related option. DHCP or static. If DHCP is selected, it indicates CPE will get the host IP address automatically from the DHCP server. Default: DHCP
netmask < <i>value</i> >	Specifies the subnet mask for IP host services.
gateway < <i>IP address</i> >	Specifies the default gateway address used for IP host services. Default: 0.0.0.0
primary-dns < <i>IP address</i> >	Specifies the primary DNS IP address. If this value is 0.0.0.0, no primary SIP DNS is defined. Default: 0.0.0.0
secondary-dns < <i>IP address</i> >	Specifies the secondary DNS IP address. If this value is 0.0.0.0, no second SIP DNS is defined. Default: d0.0.0.0
igmp-function < none snooping proxy snoopingproxy >	Enable IGMP function option. Default: none
firewall-access < http ping snmp snmptrap ssh telnet all none >	Lists the protocols allowed on this interface. The firewall option in the CPE system common profile must be enabled before these settings will take effect.
dns-type < default static proxy >	Specifies the DNS type: Default - Get the DNS information from the WAN interface Static - The DNS information is manually provisioned Proxy - Enable interface to act as a proxy for DNS requests Default: default

The following example creates a static CPE IP common profile for voice services:

- 1 Create a CPE IP common profile with profile-name. The profile index will be generated automatically.

```
zSH> CPE> RG> LAN> IP-COM> add LANIPserver host-ip-option static netmask
255.255.255.0 gateway 172.168.3.254 primary-dns 172.168.19.1
Profile "IPserver" has been created with index 3
```

- 2 Show the default CPE IP common profiles and user-created CPE IP common profiles.

```
zSH> CPE> RG> LAN> IP-COM> show all
```

				host IP			
secure	default			option	gateway	igmp fn	nat
Index	Profile Name						
fw	iface	dns src	dns type				
=====	=====	=====	=====	=====	=====	=====	=====
1	Default_Cpe_Ip_Server			dhcp	0.0.0.0	none	nat
disabl	false	false	default				
2	IPserver			static	172.168.3.254	none	nat
disabl	false	false	default				
3	LANIPserver			static	172.168.3.254	none	nat
disabl	false	false	default				
100001	default-lan-ip-server100001			static	0.0.0.0	none	nat
disabl	false	false	default				

4 entries found.

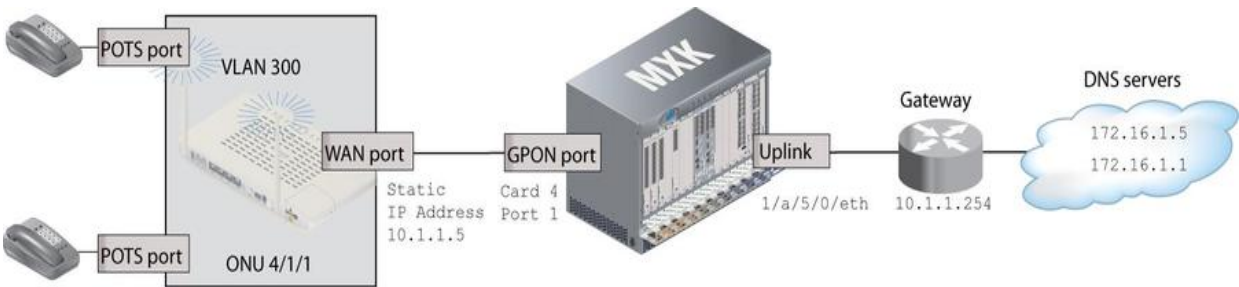
- 3 If users want to delete a user-created CPE IP common, use the delete command with the profile index or profile name.

```
zSH> CPE> IP> IP-COM> delete LANIPserver
Profile has been deleted.
```

Static configuration on the WAN side interfaces (without DHCP)

This section describes how to assign a static IP address to the WAN side interfaces on a zNID.

Figure 120: Static configuration on the WAN interfaces



Assigning a Static IP address to the WAN side interface on the zNID

This example is the continuance of the triple-play services that created in the section [OMCI GPON zNID with RG Features Installation for Triple Services on page 799](#).

To assign a static IP address to the WAN side interface, use the following procedure:

- 1 Create services on one ONT with the **bridge add** command (using system defaults).

Refer to [OMCI GPON zNID with RG Features Installation for Triple Services on page 799](#).

- 2 Verify the created services on the ONT:

```
zSH> bridge show
```

Type	Orig VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn		Tagged 100	1/4/1/1/gpononu	1-4-1-303-gponport-100/bridge	UP	D 00:02:71:19:4b:28
dwn		Tagged 200	1/4/1/1/gpononu	1-4-1-403-gponport-200/bridge	UP	
dwn		Tagged 300	1/4/1/1/gpononu	1-4-1-501-gponport-300/bridge	UP	D 00:02:71:19:4b:29 D 00:02:71:19:4b:28
upl		Tagged 100	1/a/5/0/eth	ethernet5-100/bridge	UP	S VLAN 100 default
upl		Tagged 200	1/a/5/0/eth	ethernet5-200/bridge	UP	S VLAN 200 default
upl		Tagged 300	1/a/5/0/eth	ethernet5-300/bridge	UP	S VLAN 300 default

6 Bridge Interfaces displayed

```
zSH> bridge show onu
```

ONU	GEM Port	ONU UNI	DSCP to	ONU COS	UNI VLAN/SLAN	OLT VLAN/SLAN	OLT G-VLAN	OLT MVR	Service	Rg-Mode
OLT Bridge						ST				
4/1/1	303	eth 1	-----	----	/----	Tagged 100	----	----	data	B-Routed
1-4-1-303-gponport-100/bridge						DWN				
4/1/1	303	wlan 1	-----	----	/----	Tagged 100	----	----	data	B-Routed
1-4-1-303-gponport-100/bridge						----				
4/1/1	403	eth 2	-----	----	/----	Tagged 200	----	----	iptv	Bridged
1-4-1-403-gponport-200/bridge						DWN				
4/1/1	501	pots	-----	----	/----	Tagged 300	----	----	sip	Bridged
1-4-1-501-gponport-300/bridge						UP				

3 Bridge Interfaces displayed
4 GPON ONU Connections displayed

- 3 Verify the default settings of the above voice connections on the WAN interface:

```
zSH> cpe> rg> wan > show 4/1/1 vlan 300
```

CPE	VLAN/SLAN	RG Mode	IP Address	Auth	Retry Interval	Ip-Com Profile	Port-Fwd List Profile
G-VLAN							
4/1/1	300/----	Bridged	dhcp	--	--	1	0

Pppoe User Id: --
1 services displayed

4 Prepare system wide static IP configuration:

```
zSH> CPE> RG> WAN> IP-COM> add static-ip-config-1 host-ip-option static gateway
10.1.1.254 netmask 255.255.255.0 primary-dns 172.16.1.5 secondary-dns 172.16.5.11
Profile "static-ip-config-1" has been created with index 3
```

5 Apply custom ip-config setting on WAN interface. This example shows how to apply new setting into an existing ip-com-profile. Users can apply the setting when creating the ip-com-profile too.

```
zSH> CPE> RG> WAN> modify 4/1/1 vlan 300 ip-com-profile static-ip-config-1 ip-addr
10.1.1.5
Service has been modified
```

```
zSH> CPE> RG> WAN> show 4/1/1 vlan 300
```

CPE	VLAN/SLAN	RG Mode	IP Address	Auth	Retry Interval	Ip-Com Profile	Port-Fwd List	Port-Fwd Profile
G-VLAN								
=====								
=====								
4/1/1	300/----	Bridged	10.1.1.5	--	--	3		0
--								

Pppoe User Id: --
1 services displayed

Static configuration on the LAN side interfaces with a new DHCP server

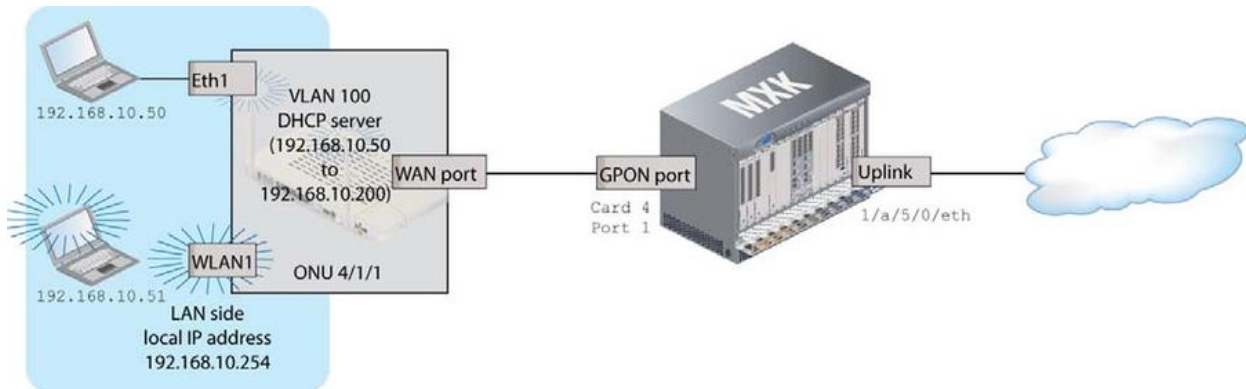
This example is the continuance of the triple-play services that created in the section [OMCI GPON zNID with RG Features Installation for Triple Services on page 799](#).

Users can set LAN side IP and DHCP server if it is in rg-brouted or rg-bpppoe.

This section configures the following settings on the Brouted LAN interfaces those has same VLAN:

- Assign a static IP address as the local IP address of the LAN interfaces
- Change the IP address range for the DHCP server
- Change the firewall access settings on the LAN interfaces

Figure 121: Static configuration on the LAN interfaces



Assigning a Static IP address to the LAN side interface on the ONU

- 1 Create services on one ONT with the bridge add command (using system defaults).

Refer to [OMCI GPON zNID with RG Features Installation for Triple Services on page 799](#).

- 2 Verify the created services on the ONT.

```
zSH> bridge show
```

Type	Orig VLAN/SLAN	VLAN/SLAN	Physical	Bridge	St	Table Data
dwn		Tagged 100	1/4/1/1/gpononu	1-4-1-303-gponport-100/bridge	UP	D 00:02:71:19:4b:28
dwn		Tagged 200	1/4/1/1/gpononu	1-4-1-403-gponport-200/bridge	UP	
dwn		Tagged 300	1/4/1/1/gpononu	1-4-1-501-gponport-300/bridge	UP	D 00:02:71:19:4b:29 D 00:02:71:19:4b:28
upl		Tagged 100	1/a/5/0/eth	ethernet5-100/bridge	UP	S VLAN 100 default
upl		Tagged 200	1/a/5/0/eth	ethernet5-200/bridge	UP	S VLAN 200 default
upl		Tagged 300	1/a/5/0/eth	ethernet5-300/bridge	UP	S VLAN 300 default

6 Bridge Interfaces displayed

```
zSH> bridge show onu
```

ONU	GEM	ONU	DSCP	ONU	UNI	OLT	OLT			
OLT Bridge	Port	UNI	to	COS	VLAN/SLAN	VLAN/SLAN	G-VLAN	MVR	Service	Rg-Mode
						ST				
4/1/1	303	eth 1	-----	----	/----	Tagged 100	----	----	data	B-Routed
1-4-1-303-gponport-100/bridge						DWN				
4/1/1	303	wlan 1	-----	----	/----	Tagged 100	----	----	data	B-Routed
1-4-1-303-gponport-100/bridge						----				
4/1/1	403	eth 2	-----	----	/----	Tagged 200	----	----	iptv	Bridged
1-4-1-403-gponport-200/bridge						DWN				
4/1/1	501	pots	-----	----	/----	Tagged 300	----	----	sip	Bridged
1-4-1-501-gponport-300/bridge						UP				

3 Bridge Interfaces displayed

4 GPON ONU Connections displayed

As shown above, data services are created on Eth 1 and WLAN 1 with brouted connections, and both ports are in VLAN 100.

3 Verify the default settings of the above data connections:

```
zSH> cpe> rg> wan > show 4/1/1 vlan 100
```

CPE	VLAN/SLAN	RG Mode	IP Address	Auth	Retry Interval	Ip-Com Profile	Port-Fwd List	Port-Fwd Profile
G-VLAN								
4/1/1	100/----	B-Routed	dhcp	--	--	1		0

--
Pppoe User Id: --
1 services displayed

```
zSH> CPE> RG> LAN> show 4/1/1 vlan 100
```

CPE	UNI	UNI-Vlan/Slan	Vlan/Slan	G-VLAN	IP-Address	IP Com Profile	Dhcp Srvr Profile
Rg-Mode							
4/1/1	eth 1		Tagged 100	----	192.168.1.1	100001	100001
B-Routed							
4/1/1	wlan 1		Tagged 100	----	192.168.1.1	100001	100001
B-Routed							

Services displayed: 2

4 Create a IP common profile:

```
zSH> CPE> RG> LAN> IP-COM> add hsi-lan host-ip-option static netmask 255.255.255.0
firewall-access all
Profile "hsi-lan" has been created with index 4
```

```
zSH> CPE> RG> LAN> IP-COM> show hsi-lan
```

Index	Profile Name	host IP option	gateway	igmp fn	nat	secure fwd	default iface	dns src	dns type
4	hsi-lan	static	0.0.0.0	none	nat	disabl	false	false	default

Net Mask: 255.255.255.0
Primary Dns: 0.0.0.0
Secondary Dns: 0.0.0.0
Firewall Access: http+ping+snmp+snmptrap+ssh+telnet
1 entries found.

5 Create a DHCP server profile:

```
zSH> CPE> RG> LAN> DHCP-SRVR> add hsi-dhcp start-addr 192.168.10.50 end-addr
192.168.10.200
Profile "hsi-dhcp" has been created with index 2
```

```
zSH> CPE> RG> LAN> DHCP-SRVR> show hsi-dhcp
```

Index	Profile Name	start addr	end addr
lease time			

=====

```

2          hsi-dhcp          192.168.10.50  192.168.10.200
-1
1 entries found.

```

- 6** Assign a new IP address, IP common profile, and a DHCP server profile to the LAN interface:

```

zSH> CPE> RG> LAN> modify 4/1/1 eth 1 vlan 100 dhcp-server-profile hsi-dhcp ip-addr
192.168.10.254 ip-com-profile hsi-lan
Service has been modified

```

As shown below, the local IP address of the BRouted LAN interfaces and the IP address range for the DHCP server are changed.

```

zSH> CPE> RG> LAN> show 4/1/1 vlan 100

```

CPE	UNI	UNI-Vlan/Slan	Vlan/Slan	G-VLAN	IP-Address	IP Com Profile	Dhcp Srvr Profile
Rg-Mode							
=====							
4/1/1 eth	1		Tagged 100	----	192.168.10.254	4	2
B-Routed							
4/1/1 wlan	1		Tagged 100	----	192.168.10.254	4	2
B-Routed							
Services displayed: 2							

Configuration of Static Routes

This section describes how to add static routes to the zNID.

To create a static route list, use the **cpe system common static-route add** command.

Command:

cpe system common static-route add <list-name>

[**dest-ip** <IP address>]

[**netmask** <netmask>]

[**gateway** <value>]

[**metric** <value>]

To create a new cpe system common static-route, the <list-name> must be provided.

[Table 88](#) provides the description for command options in the **cpe system common static-route add** command.

Table 88: CPE system common static-route add Command Options Explanation

Command Option	Description
<i>list-name</i>	The list name of the static route.

Table 88: CPE system common static-route add Command Options Explanation

Command Option	Description
dest-ip value	The IP address of the destination network or host. Host portion of the destination address must be zero. Default: 0.0.0.0
netmask value	Destination subnet mask. An value of 0.0.0.0 indicates no destination subnet mask is specified. Default: 255.255.255.0
gateway value	Next hop IP address. The next hop must be reachable. Default: 0.0.0.0
metric value	Number of hops to reach destination. A value of 0 indicates this metric is not used. Default: 1 Values: 0 - 2147483647

Adding a Static Route to the ONU

- 1 Create services on one ONU (using system defaults) with the **bridge add** command:

Refer to [OMCI GPON zNID with RG Features Installation for Triple Services on page 799](#)

- 2 Add a static route.

```
zSH> CPE> SYSTEM> COMMON> STATIC-ROUTE> add video-route dest-ip 10.2.1.0 netmask 255.255.255.0 gateway 10.1.1.253 metric 1
```

Profile "video-route" has been created with index 1/1

```
zSH> CPE> SYSTEM> COMMON> STATIC-ROUTE> show all
```

Static-Route-List profile : video-route (1)

ListIndex/

EntryIndex	Dest Ip	Netmask	Gateway	Metric
1/1	10.2.1.0	255.255.255.0	10.1.1.253	1

1 entries found.

- 3 Associate this static route with the CPE system common profile:

```
zSH> CPE> SYSTEM> COMMON> add mmr static-route-list-profile video-route
```

Cpe System Common profile "mmr" has been created with index 3

```
zSH> CPE> SYSTEM> COMMON> show mmr
```

Index	Profile Name	Firewall	Sync Cookie Protection	Cross VLAN Routing	Static Route List
3	video-route/1	mmr enabled	enabled	disabled	

```

Acs URL      :
Acs Username :
1 entries found.

```

- 4 The cross VLAN routing is disabled by default. This example enables the crossing VLAN routing on the static route.
 - If "enabled" is selected for cross VLAN routing, routing between VLANs is allowed. Route table lookups ignore the VLAN ID of the ingress and egress ports. If there is a match, the packet is routed out the interface specified in the Route table, regardless of which VLAN it is a member of.
 - If "disabled" is selected for cross VLAN routing, packets will be forwarded to the configured Default Route for the VLAN that they arrived on, unless there is a Route Table match within that same VLAN. Routing of packets across VLANs is prevented, providing traffic isolation.

```

zSH> CPE> SYSTEM> COMMON> modify mmr cross-vlan-routing enabled
Cpe System Common profile has been modified

```

```

zSH> CPE> SYSTEM> COMMON> show mmr

```

Index	Profile Name	Firewall	Sync Cookie Protection	Cross VLAN Routing	Static Route List
3	mmr	enabled	enabled	enabled	video-route/

```

1
Acs URL      :
Acs Username :
1 entries found.

```

- 5 Apply static route to a ONU.

If this CPE does not have system common profile assigned to it, users can use the **cpe system add** command. Users can also change the system common profile on the CPE by using the **cpe system modify** command.

```

zSH> CPE> SYSTEM> add 4/1/1 sys-common-profile mmr
System profile "4/1/1" has been created

```

```

zSH> CPE> SYSTEM> show 4/1/1
CPE      SYSTEM COMMON PROFILE
=====
4/1/1    mmr/3
1 entries found.

```

Configuration of DNS Hosts and DNS Proxy

This section describes how to add DNS hosts to the ONT. Note that only after enabling DNS proxy, the DNS host will be used on the ONU.

When DNS Proxy is selected as the DNS Relay Source on a LAN-side interface, client devices will send all DNS requests to this LAN side IP Address. This Router will check the local DNS Host Table for any pre-configured domain name lookups, and if a matching entry is found, it will respond with the corresponding IP Address. When there are no matching entries in the local Host Table, this router will initiate a proxy DNS request using its system DNS source. It will then generate a corresponding DNS response to the LAN-side client with the corresponding IP Address learned via the proxy request.

To create a DNS host list, use the **cpe system common dns-host add** command.

Command:

cpe system common dns-host add < list-name>

[**domain-name** < domain-name >]

[**ip-address** < ip-address>]

To create a new cpe system common dns-host, the <list-name> must be provided.

[Table 89](#) provides the description for command options in the **cpe system common dns-host add** command.

Table 89: CPE system common dns-host add Command Options Explanation

Command Option	Description
<i>list-name</i>	The list name of the DNS host.
domain-name value	The domain name assigned to the host IP address.
ip-address value	Host IP address.

Adding a DNS host list to the ONU and Enabling DNS proxy on ONU LAN ports

- 1 Create services on one ONU (using system defaults) with the **bridge add** command:

Refer to [OMCI GPON zNID with RG Features Installation for Triple Services on page 799](#)

- 2 Add a DNS host list.

```
zSH> CPE> SYSTEM> COMMON> DNS-HOST> add DNSTest domain-name zhong.com ip-address 199.190.211.11
```

```
cpe-dns-host-list profile "DNSTest" with index 1 has been created
cpe-dns-host profile 1/1 has been created in list "DNSTest"
```

```
zSH> CPE> SYSTEM> COMMON> DNS-HOST> add DNSTest domain-name
discovery.ip.tv.microsoft.com ip-address 10.10.10.1
```

cpe-dns-host profile 1/2 has been created in list "DNSTest"

```
zSH> CPE> SYSTEM> COMMON> DNS-HOST> show DNSTest
Dns-Host-List profile : DNSTest (1)
ListIndex/
EntryIndex   IP Address           Domain Name
=====
      1/1   199.190.211.11    zhong.com
      1/2     10.10.10.1    discovery.iptv.microsoft.com
2 entries found.
```

3 Associate this DNS host list with the CPE system common profile.

This example adds a new CPE system common profile CommonTest and specifies the dns-host-list-profile is DNSTest. Users can also modify an existing CPE system common profile to add the dns-host-list-profile, if users wish to apply this dns-host-list to multiple ONUs.

```
zSH> CPE> SYSTEM> COMMON> add CommonTest dns-host-list-profile DNSTest
Cpe System Common profile has been modified
```

```
zSH> CPE> SYSTEM> COMMON> show CommonTest
Index      Profile Name      Firewall  Sync Cookie  Cross VLAN  Static Rte  Dns Host  TR69  Inform
=====
      3      CommonTest  enabled   enabled     disabled    0           0         1     enabled   300
Acs URL      :
Acs Username : admin
1 entries found.
```

4 Associate DNS host list to a ONU.

If this CPE does not have system common profile assigned to it, users can use the **cpe system add** command. Users can also change the system common profile on the CPE by using the **cpe system modify** command.

```
zSH> CPE> SYSTEM> add 4/1/1 sys-common-profile CommonTest
System profile "4/1/1" has been created
```

```
zSH> CPE> SYSTEM> show 4/1/1
CPE      SYSTEM COMMON PROFILE
=====
      4/1/1      CommonTest/3
1 entries found.
```

5 In order to use the DNS host list on the ONU LAN ports, the DNS type must be set to proxy.

- If "Default" is selected for DNS type, LAN interface will get the DNS information from the WAN interface. This is the default value.
- If "Static" is selected for DNS type, the DNS information is manually provisioned.
- If "Proxy" is selected for DNS type, the LAN interface will be enabled to act as a proxy for DNS request.

- a** Show the IP common profiles assigned to the CPE LAN ports on the ONU.

The following two examples show the IP common profile assigned on ONU 4/1/1 Ethernet port 1 is 100001, and DNS type is set to default.

```
zSH> CPE> RG> LAN> show 4/1/1
```

CPE	UNI	UNI-Vlan/Slan	Vlan/Slan	G-VLAN	IP-Address	IP Com Profile	Dhcp Srvr Profile	Rg-Mode
4/1/1	eth 1	1	0,100/----	----	192.168.1.1	100001	100001	B-Routed
4/1/1	eth 2	2	0,200/----	----	0.0.0.0	0	0	Bridged
4/1/1	pots		0,300/----		---	---	---	Bridged

Services displayed: 3

```
zSH> CPE> RG> LAN> IP-COM> show 100001
```

Index	Profile Name	host IP option	gateway	igmp fn	nat	secure fwd	default iface	dns src	dns type
100001	default-lan-ip-server100001	static	0.0.0.0	none	nat	disabl	false	false	default

Net Mask: 255.255.255.0
 Primary Dns: 0.0.0.0
 Secondary Dns: 0.0.0.0
 Firewall Access: http+ping+snmp+snmptrap+ssh+telnet
 1 entries found.

- b** Modify the DNS-Type setting to Proxy in the IP common profile.

```
zSH> CPE> RG> LAN> IP-COM> modify 100001 dns-type proxy
```

Saving this profile will trigger a partial reconfiguration on each of the CPEs that depend on this profile and may cause service interruptions on those CPEs

Do you want to save the profile? [yes] or [no]: **yes**

Do you want to exit from this request? [yes] or [no]: **no**

Are you sure? [yes] or [no]: **yes**

Profile has been modified.

- c** Verify the DNS-TYPE is set to proxy now.

```
zSH> CPE> RG> LAN> IP-COM> show 100001
```

Index	Profile Name	host IP option	gateway	igmp fn	nat	secure fwd	default iface	dns src	dns type
100001	default-lan-ip-server100001	static	0.0.0.0	none	nat	disabl	false	false	proxy

Net Mask: 255.255.255.0
 Primary Dns: 0.0.0.0
 Secondary Dns: 0.0.0.0
 Firewall Access: http+ping+snmp+snmptrap+ssh+telnet
 1 entries found.

Configuration of Firewall

User can enable or disable firewall on the CPE. Enabling firewall can protect the CPE from unwanted instruction. When firewall is enabled, incoming connections can still be selectively allowed through firewall access and port forwarding settings. The firewall is enabled by default.

Enabling or disabling firewall

Use the CPE system common profile to enable or disable firewall settings. By default, the firewall is enabled.

1 Verify the firewall setting in the CPE:

```
zSH> CPE> SYSTEM> show 3/4/1
CPE      SYSTEM COMMON PROFILE
=====
      3/4/1                2/2
1 entries found.
```

```
zSH> CPE> SYSTEM> COMMON> show 2
```

Index	Profile Name	Firewall	Sync Cookie Protection	Cross VLAN Static Route Routing	List
2		2 enabled	enabled	disabled	None/ 0

```
Acs URL      :
Acs Username :
1 entries found.
```

2 Modify firewall setting:

```
zSH> CPE> SYSTEM> COMMON> modify 2 firewall disabled
Cpe System Common profile has been modified
```

Configuring firewall access

Firewall access control manages the protocols allowed on the CPE WAN or LAN interfaces. It requires firewall feature to be enabled.

The protocols are listed below:

- HTTP: Web Browser Traffic.
- PING: ICMP Echoes used to test for connectivity.
- SNMP: Simple Network Management Protocol.
- SNMPTRAP: Alarms for Simple Network Management Protocol.
- SSH: Secure Shell.
- TELNET: Remote Terminal support.

By default, those protocol are all enabled on the CPE LAN interfaces. Only PING is enabled on the CPE WAN interfaces. You can use the **cpe rg wan/lan ip-com add/modify** command to set the firewall access settings on WAN or LAN interfaces.



Note: Modifying a CPE IP Common profile will trigger a partial reconfiguration on each of the CPEs that depend on this profile and may cause service interruptions on these CPEs.

To modify the default firewall access settings on WAN or LAN interfaces, use the following procedure:

- 1** Make sure the CPE has the firewall setting enabled.
Refer to Section [Enabling or disabling firewall](#), page 845.
- 2** Change the firewall access settings on the CPE WAN interface:
 - a** Show the default firewall access setting (ping) on the CPE WAN interface:

```
zSH> CPE> RG> WAN> show 3/4/1
```

CPE Profile	VLAN/SLAN	RG Mode	IP Address	Auth	Retry Interval	Ip-Com Profile	Port-Fwd List
3/4/1	102/----	B-Routed	dhcp	--	--	1	0

--
Pppoe User Id: --
1 services displayed

```
zSH> CPE> RG> WAN> ip-com show 1
```

Index	Profile Name	option	gateway	igmp fn	nat
1	Default_Cpe_Ip_Server	dhcp	0.0.0.0	none	nat

enabl ping
Net Mask: 255.255.255.0
Primary Dns: 0.0.0.0
Secondary Dns: 0.0.0.0
1 entries found.

- b** Create a new CPE IP common profile with firewall-access settings set to ping and HTTP:

```
zSH> CPE> RG> WAN> IP-COM> add FWHTTP firewall-access ping HTTP
Profile "FWHTTP" has been created with index 2
```

- c** Apply the new firewall access settings to the WAN interface:

```
zSH> CPE> RG> WAN> modify 3/4/1 ip-com-profile FWHTTP
Service has been modified
```

- 3** Change the firewall access settings on the CPE LAN interface:
 - a** Show the default firewall access setting (http+ping+snmp+snmptrap+ssh+telnet) on the CPE LAN interface:

```
zSH> CPE> RG> LAN> show 3/4/1 eth 1
```

CPE	UNI	UNI-Vlan/Slan	Vlan/Slan	G-VLAN	IP-Address	IP Com Profile	Dhcp Srvr Profile	Rg-Mode

```

=====
3/4/1 eth 1          Tagged 102 ---- 192.168.1.1 100001 100001 B-Routed
Services displayed: 1

```

```
zSH> CPE> RG> LAN> ip-com show 100001
```

```

      host IP
Index  Profile Name      option  gateway  igmp fn  nat  secure  firewall-access
=====
100001 default-lan-ip-server100001 static    0.0.0.0   none   nat  disabl
http+ping+snmp+snmptrap+ssh+telnet
Net Mask: 255.255.255.0
Primary Dns: 0.0.0.0
Secondary Dns: 0.0.0.0
1 entries found.

```

b Apply the new firewall access settings to the LAN interface:

```
zSH> CPE> RG> LAN> modify 3/4/1 ip-com-profile FWHTTP
```

Saving this profile will trigger a partial reconfiguration on each of the CPEs that depend on this profile and may cause service interruptions on those CPEs

Do you want to save the profile? [yes] or [no]: **yes**

Do you want to exit from this request? [yes] or [no]: **no**

Are you sure? [yes] or [no]: **yes**

Service has been modified

```
zSH> CPE> RG> LAN> show 3/4/1 eth 1
```

```

      IP Com  Dhcp Srvr
CPE  UNI    UNI-Vlan/Slan  Vlan/Slan  G-VLAN  IP-Address  Profile  Profile  Rg-Mode
=====
3/4/1 eth 1          Tagged 102 ---- 192.168.1.1 2 100001  B-Routed
Services displayed: 1

```

Configuring port forwarding

The CPE port forwarding list reflects the existing port forwarding rules. The port forwarding list only can be created on a WAN interface that has NAT or NAPT enabled. By default, there is no port-forwarding list associated to the WAN interface.

Port forwarding list requires firewall feature to be enabled.

To create CPE port forwarding list, use the **cpe rg wan port-fwd add** command.

[Table 90](#) provides the description for command options in the **cpe rg wan port-fwd add** command.

Table 90: cpe rg wan port-fwd add Command Options Explanation

Command Option	Description
<i>list-name</i>	The list name of the port forwarding rules.

Table 90: cpe rg wan port-fwd add Command Options Explanation

Command Option	Description
type <dmz portrange portremap>	Type of the port forwarding. Default: dmz Values: dmz When DMZ is chosen it is the only rule allowed on that interface. A DMZ rule is effectively the same as a rule with all ports included. Range rules are more secure than setting a DMZ rule, because Range rules allow ports or groups of ports to be opened up. portrange Range indicates that any traffic on those ports will be sent to the private IP address. portremap Remap indicates that any traffic on those ports will be sent to the private IP address at the private port.
start-port <i>PortNumber</i>	Lowest value port number for the range. Default: 0 Values: 0-65535
end-port <i>PortNumber</i>	Highest value port number for the range. This can be equal to port-start if there is only one port. Default: 0 Values: 0-65535, end-port must be larger or equal to start-port.
protocol <none tcp udp tcpudp icmp icmpv4>	Indicate which protocols to monitor for the port numbers. Default: none Values: tcp, udp, tcp-udp, icmp, icmpv4, none.
private-port <i>PortNumber</i>	The port number with which to send the traffic. Default: 0 Values: 0-65535
private-ip <i>IPaddress</i>	The port IP Address with which to send the traffic. Default: 0.0.0.0

To configure port forwarding on CPE, use the following procedure:

- 1 Make sure the CPE has the firewall setting enabled.
Refer to Section [Enabling or disabling firewall, page 845](#).
- 2 Create a port forwarding list and add two entries into it:

```
zSH> CPE> RG> WAN> PORT-FWD> add port-fwd-1 type portrange start-port 80 end-port
81 protocol udp private-ip 192.168.10.2
Profile "port-fwd-1" has been created with index 1/1
```

```
zSH> CPE> RG> WAN> PORT-FWD> add port-fwd-1 type portrange start-port 5500
end-port 5500 protocol udp private-ip 192.168.10.2
Profile "port-fwd-1" has been created with index 1/2
```

```
zSH> CPE> RG> WAN> PORT-FWD> show all
Port-Fwd-List profile : port-fwd-1 (1)
ListIndex/      Start End      Private      Private
EntryIndex  Type      Port  Port  Protocol  Port      Ip
=====
      1/1 portrange   80    81    udp        0    192.168.10.2
      1/2 portrange 5500  5500    udp        0    192.168.10.2
2 entries found.
```

- 3 Associate the WAN interface with the port forwarding list. Note that the VLAN ID must be specified

```
zSH> CPE> RG> WAN> modify 3/4/1 vlan 102 port-fwd-list-profile 1
Service has been modified
```

Configuration of DHCP Server

DHCP server configuration is only for rg-brouted or rg-bpppoe mode connection. The rg-bridged mode connection does not need it.

After creating the connection between the MXK GEM port and CPE UNI port with the **bridge add** command, a default DHCP server with index 100001 is automatically created for the LAN side devices.

The default DHCP server has the following settings:

profile-name= default-dhcp-server10001

start-addr= 192.168.1.10

end-addr= 192.168.1.100

lease-time= 86400 seconds

If users don't want to use the default DHCP server, users can create a new DHCP server for the LAN-side devices with the following procedure.

Creating a new DHCP server for the LAN interface

- 1 Create the LAN-side interfaces in RG for the RG-brouted connection.

```
zSH> bridge add 1-1-3-2/gpononu gem 302 gtp 1 downlink tagged vlan 102 eth 1
rg-brouted
```

- 2 Create a new DHCP server profile for the LAN-side devices:

```
zSH> CPE> RG> LAN> DHCP-SERVER> add example-dhcp start-addr 192.168.10.10 end-addr
192.168.10.100
Profile "example-dhcp" has been created with index 1
```

```
zSH> CPE> RG> LAN> DHCP-SRVR> show all
      Index      Profile Name      start addr      end addr      lease time
```

```
=====
=====
1          example-dhcp          192.168.10.10  192.168.10.100  -1
100001     default-dhcp-server100001  192.168.1.10  192.168.1.100  86400
2 entries found.
```

3 Use the new DHCP server profile in the brouted RG connection:

```
zSH> CPE> RG> LAN> modify 1/3/2 eth 1 vlan 102 dhcp-server-profile example-dhcp
ip-addr 192.168.10.1
```

```
zSH> CPE> RG> LAN> show 1/3/2 vlan 102
```

CPE	UNI	UNI-Vlan/Slan	Vlan/Slan	G-VLAN	IP-Address	IP Com Profile	Dhcp Srvr Profile	Rg-Mode
1/3/2 eth	1		Tagged 102	----	192.168.10.1	100001		1 B-Routed

Services displayed: 1

Configuration of Conditional DHCP server

Same as DHCP server configuration, Conditional DHCP server configuration is only for rg-brouted or rg-bpppoe mode connection. Conditional DHCP addressing is used to assign permanent IP address to Set Top Box (STB) and Digital Video Recorder (DVR) devices based on Vendor Class Identifier (VCI) or Organizationally Unique Identifier (OUI) classification. These devices are assigned IP addresses from a dedicated range within the subnet.

Users can create Conditional DHCP server with **cond-dhcp add** command and then associate it with ONU's DHCP server.

The settings of the conditional DHCP server is listed in [Table 91](#).

Table 91: cpe rg lan dhcp-srvr cond-dhcp add Command Options Explanation

Command Option	Description
<i>list-profile-name</i>	The list name of the conditional DHCP server.
admin-state <up down>	Enable/disable this conditional DHCP server entry. Disabling allows this conditional DHCP server entry to be temporarily unavailable for testing or debug purposes. Default: enabled
vci-oui VCI or OUI	This field is for Vendor Class Identifier (Option 60) or Organizationally Unique Identifier. Conditional DHCP Rules classify ingress packets based on a matching OUI, or based on a matching DHCP Option 60 string value. OUI is the first 3 bytes of a MAC, separated by colons (e.g. 00:02:71) VCI is the text string included in the Option 60. When an attached device sends a DHCP Discover packet to request a dynamically assigned IP address, it may include this Option 60 string to identify what type of device it is.

Table 91: cpe rg lan dhcp-srvr cond-dhcp add Command Options Explanation

Command Option	Description
vci-match <prefix suffix substring>	VCI, the Option 60 String included in the DHCP Discover packet, may be up to 255 characters long, while the vci-oui field supports a maximum of 32 characters. This configuration option specifies where to look in the entire Option 60 string for a match. Choices are at the beginning (Prefix), at the end (Suffix), or anywhere in the string (Substring). Default: prefix
start-addr IPAddress	Starting IP Address for the subnet range associated with this rule. Must be in the same subnet as the LAN IP Interface for this VLAN. Every Conditional DHCP Rule should have a unique range of IP Addresses within the LAN Subnet (non-overlapping).
end-addr IPAddress	Last IP Address associated with this rule.
wan-vlan VLANID	VLAN ID of the WAN interface that shall be used as the default forwarding path for all packets received with an IP address in this subnet. This feature enables automatic forwarding of packets sourced by a Conditional DHCP Client device to a different Default Gateway (based on the VLAN ID).
port-fwd-rule <enabled disabled>	When enabled, a Static Port Forwarding Firewall Rule will automatically be created for each client device assigned an IP Address based on this rule. Default: disabled
start-port PortNumber	WAN-side Port Number of the Port Forwarding Rule that will be automatically created for the Conditional DHCP client assigned the Starting IP Address defined for this rule. The WAN-side Port Number will increment by 1 for each new client device. Default: 65501
private-port PortNumber	The LAN-side Port Number of the Port Forwarding Rule that will be bound to the private IP Address assigned to each client. This Port Number is typically assigned based on the application that the Port Forwarding Rule will be used for. For example, if the Rule will be used to enable on-demand SSH connections to client devices with conditional IP Addresses, then the Private Port will be 22. Default: 0
protocol < tcp udp both >	Defines what type of packets will be allowed through the Static Port Forwarding Rule Default: tcp
pri-dns IPAddress	Defines the Option 6 Primary DNS IP Addresses that will be included in the DHCP Offer to Conditional DHCP client devices matching this rule.
sec-dns IPAddress	Defines the Option 6 Secondary DNS IP Addresses that will be included in the DHCP Offer to Conditional DHCP client devices matching this rule.
vsi String	Vendor Specific Information (Option 43) is a user-configured text string of up to 32 characters with no restrictions on format or content. Required by some DHCP client devices for specific applications and/or boot up procedure.
permanent < enabled disabled >	When enabled, IP Addresses are permanently bound to the MAC address of the device they are assigned to. DHCP Lease table will show permanent entries for each host. Default: enabled

Creating a conditional DHCP server for the STB or DVR

- 1 Create the LAN-side interfaces in RG for the RG-brouted connection.

```
zSH> bridge add 1-2-1-1/gpononu gem 301 gtp 1 downlink vlan 1000 tagged eth 1
rg-brouted
```

Adding bridge on 1-2-1-1/gpononu

Created bridge-interface-record 1-2-1-301-gponport-1000/bridge

CPE Connection 1-2-1-301/gponport/1/1/0/0 has been created

- 2 Create new conditional DHCP servers for the LAN-side STB or DVR devices.

The following examples create two new conditional DHCP server entries in the same list:

```
zSH> CPE> RG> LAN> DHCP-SRVR> COND-DHCP> add test admin-state up vci-oui 01:02:03
start-addr 192.168.1.200 pri-dns 172.16.1.50 end-addr 192.168.1.220
Profile "test"/1 has been created
```

```
zSH> CPE> RG> LAN> DHCP-SRVR> COND-DHCP> add test admin-state up vci-oui 01:01:01
start-addr 192.168.1.100 pri-dns 192.168.1.130 end-addr 192.168.1.120
Profile "test"/2 has been created
```

```
zSH> CPE> RG> LAN> DHCP-SRVR> COND-DHCP> show test
```

Index	Profile Name	VCI/OUi	vci-match	start addr	end addr	primary dns
1/1	test	01:02:03	prefix	192.168.1.200	192.168.1.220	172.16.1.50
0.0.0.0						
Admin State	: up					
WAN VLAN	: 0					
Port Forwarding Rule	: disabled					
Start Port	: 65501					
Private Port	: 0					
Protocol	: tcp					
Vendor Specific Info	:					
Permanent	: enabled					

Index	Profile Name	VCI/OUi	vci-match	start addr	end addr	primary dns
1/2	.	01:01:01	prefix	192.168.1.100	192.168.1.120	192.168.1.130
0.0.0.0						
Admin State	: up					
WAN VLAN	: 0					
Port Forwarding Rule	: disabled					
Start Port	: 65501					
Private Port	: 0					
Protocol	: tcp					
Vendor Specific Info	:					
Permanent	: enabled					

2 entries found.

- 3 Associate the new conditional DHCP server list to the DHCP server profile.

Show the DHCP server profile ID of ONU 2/1/1.

```
zSH> CPE> RG> LAN> show 2/1/1
```

IP Com Dhcp Srvr


```

CPE  UNI    UNI-Vlan/Slan  Vlan/Slan  G-VLAN  IP-Address  Profile  Profile
Rg-Mode
=====
2/1/1 eth  1              0,1000/----  ----    192.168.1.1  100001  100001
B-Routed
Services displayed: 1

```

Associate conditional DHCP server list to the DHCP server.

```

zSH> CPE> RG> LAN> DHCP-SRVR> modify 100001 cond-dhcp-srv-list test
Profile has been modified.

```

```

zSH> CPE> RG> LAN> DHCP-SRVR> show all
Index          Profile Name          start addr      end addr
lease time     cond DHCP list
=====
100001         default-dhcp-server100001  192.168.1.10   192.168.1.100
86400          1
1 entries found.

```

- 4 Note that the Primary and Secondary NTP servers for a Conditional DHCP Server is from ntp-client-config 0 profile.

```

zSH> get ntp-client-config 0
ntp-client-config 0
primary-ntp-server-ip-address: ---> {128.2.136.71} Primary NTP Server
secondary-ntp-server-ip-address: -> {50.31.2.213} Secondary NTP Server
local-timezone: -----> {gmt}
daylight-savings-time: -----> {false}
daylight-savings-time-start: -----> {}
daylight-savings-time-end: -----> {}

```

Configuration of PPPoE username and password

The Point-to-Point Protocol over Ethernet (PPPoE) encapsulates PPP frames inside Ethernet frames to create a PPPoE tunnel between hosts connected to the zNID and other devices out in the cloud. While Ethernet is packet-based (so no direct connection is opened), PPP is a direct connection where one device directly connects to another using the protocol. PPPoE is a virtual connection (usually called tunnel) between two devices.

By specifying **rg-bpppoe** mode in the **bridge add** command, users can add a PPPoE on a Uni-port by VLAN. PPPoE/Bridged VLANs are similar to Brouted VLANs, but the WAN side interface is a PPPoE client that establishes a PPPoE tunnel to an upstream BRAS. On the LAN side of a PPPoE/Bridged VLAN, all ports will be members of the same IP subnet.

The configuration of the PPPoE session includes the following items:

- PPPoE user name (required)
- PPPoE password (required)

- PPPoE authentication method (optional, it has default value)
- PPPoE retry interval (optional, it has default value)

To configure the above items, use the **cpe rg wan modify** command.

[Table 92](#) provides the description for PPPoE related command options in the **cpe rg wan modify** command.

Table 92: cpe rg wan modify - PPPoE Related Command Options Explanation

Command Option	Description
pppoe-auth <auto pap chap mschap>	Indicates the PPP authentication protocol to be used for PPPoE authentication. Default: auto Values: auto pap chap mschap
pppoe-retry-interval <i>value</i>	Specifies the time in seconds before retrying connection. Default: 3 Values: from 1 to 2147483647 seconds
pppoe-usr-id <i>value</i>	The login user name to be used for PPPoE authentication. Values: an unique 25-char string
pppoe-password <i>value</i>	The login password to be used for PPPoE authentication. Values: an encrypted 25-char string

Specifying a PPPoE username and password

- 1 Create the LAN-side interfaces in RG for the RG-bpppoe connection.

```
zSH> bridge add 1-1-3-2/gpononu gem 302 gtp 1 downlink tagged vlan 102 eth 1
rg-bpppoe
Adding bridge on 1-1-3-2/gpononu
Created bridge-interface-record 1-1-3-302-gponport-102/bridge
CPE Connection 1-1-3-302/gponport/1/2/0/0 has been created
```

- 2 Configure PPPoE user name and password on the CPE RG WAN interface on the zNID.

```
zSH> CPE> RG> WAN> modify 1/3/2 vlan 102 pppoe-usr-id smith pppoe-password
zhone1234
Service has been modified
```

```
zSH> CPE> RG> WAN> show 1/3/2 vlan 102
```

Retry Ip-Com Port-Fwd

```

CPE  VLAN/SLAN  RG Mode  IP Address  Auth  Interval  Profile List  Profile
G-VLAN
=====
1/3/2 102/---- B-PPPoE      dhcp    auto      3         1           0          --
Pppoe User Id: smith
1 services displayed

```

Configuration of TR-069

TR-069(Technical Report 069) is a management protocol which allows an Auto-Configuration Server (ACS) to auto-configure, provision, collection, and provide diagnostics to the zNID.

Creating TR-069 connection and configuring TR-069 client options

The following procedure shows how to create TR-069 connection and configure the TR-069 client.

- 1 Create a TR-069 connection on ONU 1/2/1 GEM port 601.

```

zSH> bridge add 1-1-2-1/gpononu gem 601 gtp 1 downlink vlan 501 tagged tr69
rg-bridged

```

Adding bridge on 1-1-2-1/gpononu

Created bridge-interface-record 1-1-2-601-gponport-501/bridge

CPE Connection 1-1-2-601/gponport/20/0/0/0 has been created

```

zSH> bridge show

```

```

Orig
Type VLAN/SLAN  VLAN/SLAN  Physical
Bridge                               St  Table Data
-----
dwn      Tagged 105    1/1/2/1/gpononu      1-1-2-301-gponport-105/
bridge   UP    D 00:02:71:18:f8:ba
dwn      Tagged 501    1/1/2/1/gpononu      1-1-2-601-gponport-501/
bridge   UP
dwn      Tagged 202    1/1/2/5/gpononu      1-1-2-305-gponport-202/
bridge   UP
dwn      Tagged 105    1/3/4/1/gpononu      1-3-4-301-gponport-105/
bridge   DWN
dwn      Tagged 105    1/3/4/1/gpononu      1-3-4-401-gponport-105/
bridge   DWN
upl      105      1/a/5/0/eth          ethernet5/
bridge   UP    S VLAN 105 default
upl      Tagged 8      1/a/8/0/eth          ethernet8-8/
bridge   DWN   S VLAN 8 default
tls      Tagged 201    1/a/8/0/eth          ethernet8-201/
bridge   DWN
8 Bridge Interfaces displayed

```

```
zSH> bridge show onu
```

ONU	GEM Port	ONU UNI	DSCP to COS	ONU UNI VLAN/SLAN	OLT VLAN/SLAN ST	G-VLAN	OLT MVR	Service	Rg-Mode
1/2/1	601	tr69	-----	----/----	Tagged 501	----	----	tr69	Bridged
1-1-2-601-gponport-501/bridge UP									
3/4/1	401	eth 2	-----	----/----	Tagged 105	----	----	data	-----
1-3-4-401-gponport-105/bridge DWN									
1/2/1	301	eth 1	-----	----/----	Tagged 105	----	----	data	B-Routed
1-1-2-301-gponport-105/bridge DWN									
1/2/5	305	pots	-----	----/----	Tagged 202	----	----	sip	Bridged
1-1-2-305-gponport-202/bridge UP									
3/4/1	301	eth 1	-----	----/----	Tagged 105	----	----	data	-----
1-3-4-301-gponport-105/bridge DWN									

5 Bridge Interfaces displayed
5 GPON ONU Connections displayed

- 2 Create a CPE system common profile for TR-069 clients, and set the ACS URL, ACS User name, and ACS Password to match the ones pre-configured on the TR-069 server (i.e.ACS):

```
zSH> CPE> SYSTEM> COMMON> add motive acs-url http://zhone.com:6050 acs-userName admin acs-password zhone
```

Cpe System Common profile "motive" has been created with index 2

```
zSH> CPE> SYSTEM> COMMON> show motive
```

Index	Profile Name	Firewall	Sync Cookie Protection	Cross VLAN Routing	Static Route List
2	motive	enabled	enabled	disabled	None/0

Acs URL : http://zhone.com:6050
Acs Username : admin
1 entries found.

- 3 Apply the new TR-069 client settings to CPE 1/2/1:

```
zSH> CPE> SYSTEM> add 1/2/1 sys-common-profile motive
```

System profile "1/2/1" has been created

```
zSH> CPE> SYSTEM> show 1/2/1
```

CPE	SYSTEM COMMON PROFILE
1/2/1	motive/2

1 entries found.

Set factory default for an ONU

Sets the ONU to the factory default. The **gpononu set2default** command causes the MXK to reset the ONU's database. After that this ONU will be automatically rebooted, and then when the ONU comes up, it will be

reconfigured with the previous configuration. This command is only applicable to the ONUs that support Residential Gateway (RG) provisioning through MXK.

It is typically only used for debug.

Set factory default for an ONU:

```
zSH> gpononu set2default 13/4/2
```

System Name and Location of zNID

The name and location of the zNID device are needed in some applications such as 911 calls. To set, modify, view, and delete those zNID name and location information, use the **cpe system sysinfo add/modify/show/delete** command.

[Table 93](#) provides the description for command options in the **cpe system sysinfo add/modify/show** command.

Table 93: cpe system sysinfo add/modify/show Command Option Explanations

Command Option	Description
<i>interfaceID</i>	ONU port ID in the format of slot ID/OLT port ID/ONU port ID.
name <32 byte character string>	Identifies the system name of the zNID device. This value will be placed on the top banner of the CPE WEBGUI screen. It could be 32-byte characters string or less. Default is blank.
location <64 byte character string>	Identifies where this zNID device resides. For example: a street address or a rack/shelf/slot description. It could be 64-byte characters string or less. Default is blank.

Setting, Modifying, Viewing, and Deleting zNID Name and Location

Note that the CPE command shell for this feature is CPE>SYSTEM>SYSINFO>.

- 1 This example sets the system name and location on ONU 13/4/3.

```
zSH> CPE> SYSTEM> SYSINFO> add 13/4/3 name zhong location oakland
Profile has been updated
```

- 1 This example modifies the location of the ONU 13/4/3.

```
zSH> CPE> SYSTEM> SYSINFO> modify 13/4/3 location oakland,california
Profile has been updated.
```

- 2 Show the system name and location of the ONU 13/4/3.

```
zSH> CPE> SYSTEM> SYSINFO> show 13/4/3
CPE      NAME      LOCATION
=====
13/4/3      zhong oakland,california
```

- 3 Delete the system name and location of the ONU 13/4/3, set them back to default.

```
zSH> CPE> SYSTEM> SYSINFO> delete 13/4/3
Profile has been deleted
```

Using Name Format of the zNID and Address Format of the zNID in GPON bridge commands

After users set the system name on the zNID, users could use the name format of the zNID and the address format in the GPON bridge related commands.

For example, if the name of the ONU 13/4/3 is “zhone”, the following commands can be used:

```
zSH> bridge add 1-13-4-3/gpononu gem 704 gtp 1 downlink
vlan 101 tagged eth 2 rg-bridged           Address format
zSH> bridge add zhone gem 704 gtp 1 downlink vlan 101
tagged eth 2 rg-bridged                   Name format
```

Guided VLAN

Guided VLAN (g-vlan) is a VLAN to be created in RG. Usually the users do not need to specify “g-vlan” keyword in a bridge command. The “vlan” parameter specified in a bridge command is also the RG VLAN. However, there are deployment scenarios that multiple services for one GPON zNID share a common VLAN in the upstream network. If users still want to isolate the services in RG, need Guided VLAN.

In the following example, HSI is tagged with VLAN 10 and VoIP is tagged with VLAN 11 in RG. The ONT inside the same zNID translates both VLAN 10 and 11 to a common VLAN 101 and promotes outer tag 1952.

```
zSH> bridge add 1-1-2-1/gpononu gem 3101 gtp 1 downlink
vlan 101 cos 4 slan 1952 tagged g-vlan 10 eth [2-4]
rg-brouted
```

```
zSH> bridge add 1-1-2-1/gpononu gem 3101 gtp 1 downlink
vlan 101 cos 5 slan 1952 tagged g-vlan 11 sip rg-bridged
```

PoE Power Control per Port & Total Power Budget

This feature is for Power Over Ethernet (PoE) application only. The Power Supply setting at the CPE system level is the rated output power in Watts of the 48V-54V power supply attached to the CPE. This value is used to determine the total PoE Power Budget.

Total PoE Power Budget is total amount of power available for all PoE devices. A single 48V-54V power input supplies power for ONT operation and powering of PoE device. To ensure that sufficient power is reserved for ONT operation, the Total PoE Power Budget will always be less than the

maximum power provided by the Power Supply. The Power Supply is configured at the System level. The Total PoE Power Budget is calculated by subtracting 20W from the Power Supply, and then rounding down to the closest multiple of 7.6W. For example, if the Power Supply is 90W, the Total PoE Power Budget = 68.4W.

```
zSH> cpe system common modify 1 power-supply 90
```

```
zSH> cpe system common show 1
```

Index	Profile Name	Firewall	Sync Cookie Protection	Cross VLAN Routing	Static Rte List	Dns Host List	TR69 Inform	Inform Interval
1	Default_Cpe_System_Common	enabled	enabled	disabled	0	0	enabled	300

```

Acs URL      :
Acs Username :
Power Supply : 90
Power Shutdown Delay : 1
Power Restore Delay : 2

```

The maximum PoE power allowed per port is defined in the Power Range field on each Ethernet port. The choices of this field are High (30.4W), Low (7.6W), Medium (15.2W), Disabled (no PoE Power will be provided).

```
zSH> cpe eth modify 6/2/1/1 power-range medium
```

```
zSH> cpe eth show all
```

CPE	Port Number	Admin	Rate	Duplex	Video Profile	Traf Mngt Profile	Alm St	Sev	Power Shed	Power Range
6/2/1	1	up	auto	auto	0	0	Dis	Mj	Dis	Medium

Power Shedding Enable/Disable Per Port

This feature is for Power Over Ethernet (PoE) application. With this feature customers can customize the services that users want to extend on battery power during power outages, for example VoIP services for emergency contact or data services on an important computer, etc., power shedding maybe used to shut down the services of the ONU.

When power shedding is enabled at the CPE system level and the CPE is operating on battery power during an AC power outage, "power-shed" controls the enable/disable state of each Ethernet port. Ports with power shed is enabled will remain operational on battery power, while disabled ports will be shut down to conserve battery power. Power shedding for the Ethernet ports are Disabled by default.

Whether power shedding in the CPE system level is enabled or disabled is controlled by the power-shutdown-delay field in the CPE system common profile. The power-shutdown-delay is 0 means disabled; the power-shutdown-delay in the range of 1 to 60 means enabled and defines the amount of time in minutes the ONU waits after an AC power outage before shut-down.

The power-restore-delay in the CPE system level determines the amount of time in minutes the ONU will wait after AC power is restored before reactivating services. The value is in the range of 0 to 10.

Enabling Power Shedding on an Ethernet Port

- 1
- This example modifies the default CPE system common profile.

It enables power shedding in the CPE system level by specifying a non-zero value in the power-shutdown-delay field, the power-shutdown-delay time is 1 minute. It specifies the power-restore-delay time is 2 minutes.

```
zSH> cpe system common modify 1 power-shutdown-delay 1 power-restore-delay 2

zSH> cpe system common show 1
```

Index	Profile Name	Firewall	Sync Cookie Protection	Cross VLAN Routing	Static Rte List	Dns Host List	TR69 Inform	Inform Interval
1	Default_Cpe_System_Common	enabled	enabled	disabled	0	0	enabled	300

AcS URL :
AcS Username :
Power Supply : 0
Power Shutdown Delay : 1
Power Restore Delay : 2

- 2
- By default, the power shedding on the ethernet ports are disabled.

This example enables power-shed on the ONU 6/2/1 Ethernet port 1.

```
zSH> cpe eth modify 6/2/1/1 power-shed enabled

zSH> cpe eth show 6/2/1/1
```

CPE	Port Number	Admin	Rate	Duplex	Video Profile	Traf Mngt Profile	Alm St	Sev	Power Shed	Power Range
2/1	1	up	auto	auto	0	0	Dis	Mj	En	Low

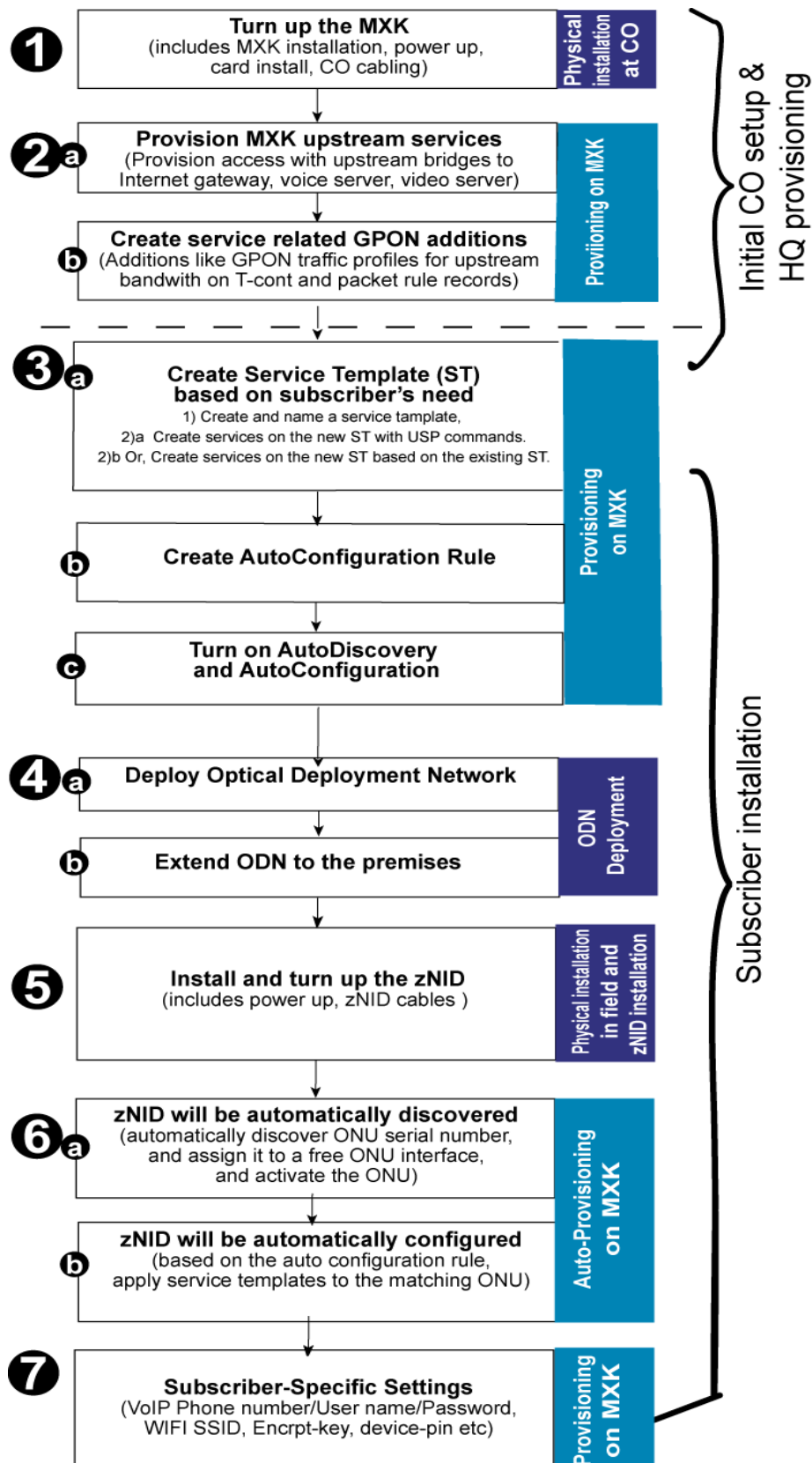
6/

AutoConfiguration and AutoDiscovery OMCI GPON zNID Installation

This section provides information on how to install OMCI-based GPON zNID with AutoConfiguration and AutoDiscovery. [Figure 122](#) shows the overall flowchart of the installation procedure. This section includes the following topics:

- [Overview, page 863](#)
- [OMCI GPON zNID installation with AutoConfiguration and AutoDiscovery for tripleplay services, page 864](#)

Figure 122: Installation procedure for OMCI GPON zNIDs with AutoConfiguration and AutoDiscovery



Overview

GPON deployments require a wide variety of skill types with varying degrees of experience. These skills types can generally be broadly characterized into two key groupings:

- Installation, and
- Configuration

Installation tasks are related to the physical racking/stacking and powering of the OLT, the deployment and termination of the fiber that connects the OLT to the ONT, and the placement of the end devices which may include but are not limited to ONUs, phone sets, and set top video boxes. While there may be other unspecified installation tasks, all are easily identifiable because they require a personnel “visit” to deploy and physically connect a collection of “boxes.” But while planning and organization are essential to a successful installation, there is only so much that can be done to drive efficiency in this brute-force part of the deployment.

On the other hand, while a completed installation is a “service enabler,” the solution configuration involves tasks that are specifically related to the customization of the installed solution to transform it into a “service aware” solution, which can be a highly customizable, can be exceedingly complex, and requires highly-skilled resource. The following features when combined together, drive deployment efficiency and accuracy for Zhone's value-added partners and enterprise customers during the configuration phase of the deployment:

Service Templates

These service templates are pre-configured profiles with a wide variety of user definable attributes like QOS, rate-limiting, LLDP, PoE, etc. Because these service templates can be archived for re-use, they can be widely deployed in "cookie-cutter" fashion in large volume but highly typical deployments like hotels, enterprise and academic campuses, and medical facilities.

AutoDiscovery

This feature ensures that when an ONU is connected for the first time to a PON link, its serial number is discovered, it is automatically assigned to a free ONU interface within the PON link, and it is activated. After the ONU is active and the OMCI channel is established, the ONU's model ID is retrieved from the ONU through OMCI. If an internal ME profile for the model exists in the system, the internal ME profile is set on the ONU.

AutoConfiguration

This feature is the automatic application of the service templates. When the ONU is first installed, this feature can work in conjunction with Auto Discovery to ensure that a default service is automatically applied to the

ONU. With additional upstream Zhone management tools, this feature can be re-used post installation to change the ONU service template to address changes in network requirements due to employee moves in enterprise environments, or possibly due to the arrival of a high-value hospitality customer who requires premium data access in his room.

OMCI GPON zNID installation with AutoConfiguration and AutoDiscovery for tripleplay services

The purpose of this section is to provision data, video, and VoIP services on the same ONU, just the MXK bridge interfaces, GEM ports, GPON traffic profiles, VLANs, UNI ports are different. For ease of discussion each of the applications is described separately in this section.

Generally these are the steps to follow to configure the MXK to be able to manage OMCI GPON zNIDs with AutoConfiguration and AutoDiscovery:

- [Create and Name a Service Template, page 864](#)
- [Create High Speed Internet Services on the Service Template, page 864](#)
- [Create Video Services on the Service Template, page 867](#)
- [Create VoIP Services on the Service Template, page 869](#)
- [View All Services Created on the Service Template, page 872](#)
- [Create AutoConfiguration Rules for the Service Template, page 873](#)
- [Turn on AutoDiscovery and AutoConfiguration in the MXK, page 875](#)
- [Verify AutoDiscovery and AutoConfiguration on ONUs, page 875](#)
- [Modify the Subscriber-Specific Settings, page 878](#)

Create and Name a Service Template

Create an empty service template with the name “tripleplay”.

```
zSH> SRVTMPL> add tripleplay
```

Template "tripleplay" has been created

Create High Speed Internet Services on the Service Template

- [Creating a GPON traffic profile for data service \(Optional\), page 865](#)
- [Creating uplink MXK bridges; adding downlink MXK bridges, CPE connections, and data services to the service template, page 865](#)
- [Performing other data related configuration \(Optional\), page 867](#)

In this example, for data services users will create uplink/downlink bridges with VLAN 102.

To create data services on the Ethernet UNI ports, use the following steps:

Creating a GPON traffic profile for data service (Optional)

Creates a GTP for high speed data configurations.

Note that the GTP is not a mandatory field in the **bridge add** command. The MXK will automatically pick one if there is no GTP IDs specified.

```
zSH> new gpon-traffic-profile 10240
gpon-traffic-profile 10241
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}:
traffic-class: -----> {ubr}:
compensated: -----> {false}:
shared: -----> {false}:
dba-enabled: -----> {false}: true
dba-fixed-us-ubr-bw: ----> {0}: 128
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: -----> {0}: 10240
dba-max-us-bw: -----> {0}: 20480
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

Creating uplink MXK bridges; adding downlink MXK bridges, CPE connections, and data services to the service template

This section will create an uplink bridges for VLAN 102 on MXK, and add data services to the “tripleplay” service template. Those data service information including adding a downlink bridge and CPE connections for the same VLAN on ONU Ethernet UNI port 1 and port 2. The following example also lists two different methods to create data services on the service template.

1 Create an uplink bridge interface on the MXK:

```
zSH> bridge add 1-a-8-0/eth uplink vlan 102
Adding bridge on 1-a-8-0/eth
Created bridge-interface-record ethernet8-102/bridge
bridge-path added successfully
```

2 Create data services on a service template— Method1:

Add data services to the new service template with the USP command, **bridge add**.

- a** This example shows ONU Ethernet UNI port 1 and 2 map to VLAN 102, and creates CPE connections on ONU Ethernet UNI port 1 and 2. As shown in the command output, the MXK automatically assigned a GEM port to the MXK bridge, both the CPE connection and the GEM port are in the same VLAN 102.

```

zSH> bridge add tripleplay gtp 10240 downlink vlan 102 tagged rg-brouted eth
[1-2]
Adding bridge on tripleplay
Created bridge-interface-record 1-0-0-257-gponport-102/bridge GEM port 257 has been
assigned automatically
CPE Connection 1-0-0-257/gponport/1/1/0/0 has been created
CPE Connection 1-0-0-257/gponport/1/2/0/0 has been created

```

b View the data service in the service template.

```

zSH> SRVTMPL> show tripleplay
CPE 0/0/3      tripleplay/gpononu
Service: DATA

```

GEM	UNI	UNI-VLAN/SLAN	VLAN/SLAN (COS,VID)	G-VLAN	Admin	Oper	Rg-Mode
257	eth 1		0,102/----	0	up		B-Routed
257	eth 2		0,102/----	0	up		B-Routed

3 Create data services on a service template— Method 2:

To create data services on a new service template based on an existing service templates, use the **srvtmpl apply** command.

With the **srvtmpl apply** command users can only change the following parameters. Users can either apply the source service template to one or multiple ONUs, or to a destination service template.

Command:

```

srvtmpl apply <src template> to <slot> [/<olt> [/<onu>]] | <dst template>
[ <uni type> <uni index> ]
[ cos <value> ] [ vlan <value> ]
[ scos <value> ] [ slan <value> ]

```

<slot>, <olt>, <onu>, and <uni index> can be specified with ranges using "[", " ", "-", or "all" (Ex: [1-3,5-7,9])

<uni type> is one of the following: eth, wlan, pots, ces

- a** As shown in the example below, by default, there are two built-in service templates: “DefaultData” and “DefaultIpPhone”.

```

zSH> SRVTMPL> list
ONU #      Name
  1      DefaultData
  2      DefaultIpPhone

```

```

zSH> SRVTMPL> show DefaultData
CPE 0/0/1      DefaultData/gpononu
Service: DATA

```

GEM	UNI	UNI-VLAN/SLAN	VLAN/SLAN (COS,VID)	G-VLAN	Admin	Oper	Rg-Mode
3000	eth 1		0,100/----	0	up		Bridged

- b** Apply the source service template “DefaultData” to the destination service template “tripleplay” on Ethernet UNI ports 1 and 2, and in VLAN 102:

```
zSH> SRVTMPL> apply DefaultData to tripleplay eth [1-2] vlan 102
Services on DefaultData are being applied to tripleplay
```

```
zSH> SRVTMPL> show tripleplay
CPE 0/0/3 tripleplay/gpononu
Service: DATA
```

GEM	UNI	UNI-VLAN/SLAN	VLAN/SLAN (COS,VID)	G-VLAN	Admin	Oper	Rg-Mode
257	eth 1		0,102/----	0	up		Bridged
257	eth 2		0,102/----	0	up		Bridged

Performing other data related configuration (Optional)

To perform the other data related configuration, such as creating CPE Ethernet subscriber profile, refer to [Create High Speed Internet on Dynamic OMCI with Uplink and Downlink](#) on page 748.

Create Video Services on the Service Template

Video bridging is very similar to data bridging, it uses downlink and uplink bridges as well, but the GTPs, GEM ports and VLANs are different.

To create video services on the Ethernet UNI ports on the same ONU, use the following steps:

- [Creating uplink and downlink bridges, and CPE connections](#), page 867
- [Performing other video related configuration \(Optional\)](#), page 868

Creating uplink and downlink bridges, and CPE connections

This section will create an uplink and downlink bridge for VLAN 203:

1 Create an uplink bridge interface

a Create the uplink bridge interface

The following example creates a video uplink bridge interface, and enables IGMP proxy with IGMP snooping.

```
zSH> bridge add 1-a-4-0/eth uplink vlan 203 tagged
igmpproxy
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-203/
bridge
Bridge-path added successfully
```

- b** Modify the bridge-path for the uplink with 30 seconds IGMP query interval. Note how the igmptimer is added to the bridge-path.

```
zSH> bridge-path modify ethernet4-203/bridge vlan
203 default igmptimer 30
Bridge-path ethernet4-203/bridge/3/203/0/0/0/0/0/0/
0 has been modified
```

2 Add video services to the “tripleplay” service template.

Create downlink bridges on the MXK and CPE connections on ONU Ethernet UNI ports.

This example maps ONU Ethernet UNI port 3 to VLAN 203, and creates CPE connections on ONU Ethernet UNI port 3. The MXK automatically assigns a GEM port and a GTP to the MXK bridge, both the CPE connection and the GEM port are in the same VLAN 203.

Users can also specify **video m/n**, *m* indicates the multicast control list on the *MXK bridge*, and *n* indicates the maximum video streams on the MXK bridge. The maximum video streams is a mandatory field when users creating a downlink bridge for video services, otherwise the bridge will fail to pass all video multicast traffic. By specifying **video 0/10** in this example users can enable subscriptions up to four video streams on the MXK bridge interface without control list checking.

If users want to control multicast control list checking on the CPE connection, use the **CPE video access add** command to create CPE video access control profiles.

This example adds VLAN 203 to the video traffic.

```
zSH> bridge add tripleplay downlink vlan 203 tagged rg-bridged eth 3 video 0/10
Adding bridge on tripleplay
Created bridge-interface-record 1-0-0-258-gponport-203/bridge      GEM port 258 has been
assigned
CPE Connection 1-0-0-258/gponport/12/3/0/0 has been created
```

3 View the data and video services added in the **tripleplay** service template.

```
zSH> srvtmpl show tripleplay
CPE 0/0/3      tripleplay/gpononu
Service: DATA
GEM  UNI  UNI-VLAN/SLAN  VLAN/SLAN(COS,VID)  G-VLAN  Admin  Oper  Rg-Mode
----  -
257  eth 1                0,102/----         0      up    B-Routed
257  eth 2                0,102/----         0      up    B-Routed
Service: IPTV
GEM  UNI  UNI VLAN/SLAN  OLT VLAN/SLAN  G-VLAN  Admin  Oper  Video Prof  Rg-Mode
----  -
258  eth 3                0,203/----         0      up                    Bridged
```

Performing other video related configuration (Optional)

To perform the other video related configurations, such as CPE video access control profile, CPE video profile, CPE ethernet subscriber profile, refer to [Create Uplink and Downlink Bridges on Dynamic OMCI for Video on page 755](#).

Create VoIP Services on the Service Template

For VoIP services Zhone recommends using uplink and downlink bridge configuration.

There are different types of VoIP signaling: SIP, SIP PLAR, H.248, or MGCP. One ONU can only use one VoIP signaling protocol.



Note: One ONU only can have one VoIP signaling. For example, if users configured POTS 1 for SIP, all the POTS ports in the same ONU must use SIP too.



Note: Make sure country code in the system profile is set properly for the voice signaling type.

To create voice services on the POTS ports on the same ONU, use the following steps:

- [Creating a GPON traffic profile \(Optional\), page 869](#)
- [Creating a CPE VoIP server profile, page 870](#)
- [Creating uplink and downlink bridges and CPE connections, page 870](#)
- [Creating a CPE IP profile for the VoIP service, page 871](#)
- [Creating a CPE VoIP subscriber profile and associate it with a VoIP server, page 871](#)
- [Performing other Voice related configuration \(Optional\), page 872](#)

Creating a GPON traffic profile (Optional)

Add the GPON traffic profile. Note that this step is not mandatory.

The following GPON traffic profile is recommended for up to four VoIP phones or four POTS ports.

```
zSH> new gpon-traffic-profile 512

gpon-traffic-profile 512
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}:
traffic-class: -----> {ubr}:
compensated: -----> {false}:
shared: -----> {false}:
dba-enabled: -----> {false}:true
dba-fixed-us-ubr-bw: ----> {0}:128
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: ----> {0}:384
dba-max-us-bw: -----> {0}:512
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved..
```

Creating a CPE VoIP server profile

This section will create a CPE VoIP server profile. Each CPE VoIP server profile specifies the IP addresses or names of the primary and secondary VoIP servers, and the VoIP signalling protocol. The VoIP signalling protocol could be SIP, SIP PLAR, H.248, or MGCP (available in RG only).



Note: The CPE VoIP server profile can only be deleted when it is not associated by any CPE VoIP subscriber profiles.

To create VoIP server profiles, use the **cpe voip server add** command.

- 1 This example creates a VoIP server profile for SIP.

```
zSH> CPE> VOIP> SERVER> add sip-test primary-server 172.16.60.51
signalling-protocol sip sip-domain metaswitch.oak.zhone.com sip-registrar
metaswitch.oak.zhone.com
Profile "sip-test" has been created with index 1
```

- 2 Show the VoIP server profile.

```
zSH> CPE> VOIP> SERVER> show sip-test
```

Index	Profile Name	Signalling Protocol	Oob Dtmf Events	Oob Cas Events	Dtmf Events Passing Method	Cas Events Passing Method
1	sip-test	sip	disabled	disabled	rfc4733	rfc4733
	Primary Server	: 172.16.60.51				
	Secondary Server	: 0.0.0.0				
	Sip Domain	: metaswitch.oak.zhone.com				
	Sip Registrar	: metaswitch.oak.zhone.com				
	Mgc Termination Id Base	:				
	Softswitch	:				
	OutBound Server	:				
	Port Id	: -1				
	Rtp Dscp	: 0				
	Signalling Dscp	: 0				
	Sip Reg Exp Time	: 3600				
	Rereg Head Start Time	: 360				
	Sip Reg Retry Time	: 60				
	Release Timer	: 10				
	Roh Timer	: 15				
	MGCP Address Mode	: ip				
	MGCP Persistent Notify	: disabled				

1 entries found.

Creating uplink and downlink bridges and CPE connections

This section will create an uplink bridge and a downlink bridge on the MXK, and create CPE connections on the ONU for the SIP service:

- 1 Create an uplink bridge on the uplink interface.

```
zSH> bridge add 1-a-5-0/eth uplink vlan 304
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-304/bridge
Bridge-path added successfully
```

- 2 Create voice services on the service template.

This example creates a downlink MXK bridge on GEM port 259 (the GEM port is automatically assigned by MXK), and creates SIP services for all POTS ports on an ONU and inserts single tagged VLAN 304 to the untagged SIP packets.

```
zSH> bridge add tripleplay gtp 512 downlink vlan 304 tagged sip rg-bridged
Adding bridge on tripleplay
Created bridge-interface-record 1-0-0-259-gponport-304/bridge Auto-assigned GEM 259
CPE Connection 1-0-0-259/gponport/14/0/0/0 has been created
```

- 3 On MXK, run the **srvtmpl show** command to view the Data, Video, Voice services that created in the “tripleplay” service template .

```
zSH> srvtmpl show tripleplay
CPE 0/0/3      tripleplay/gpononu
Service: DATA
GEM  UNI  UNI-VLAN/SLAN  VLAN/SLAN(COS,VID)  G-VLAN Admin Oper  Rg-Mode
----
257  eth 1                0,102/----         0    up    B-Routed
257  eth 2                0,102/----         0    up    B-Routed
Service: IPTV
GEM  UNI  UNI VLAN/SLAN  OLT VLAN/SLAN  G-VLAN Admin Oper  Video Prof Rg-Mode
----
258  eth 3                0,203/----         0    up                Bridged
Service: SIP
GEM  UNI  UNI VLAN/SLAN  OLT VLAN/SLAN  G-VLAN  Host IP      IP Srvr Prof
----
259  pots                0,304/----         0
```

Creating a CPE IP profile for the VoIP service

Create a CPE IP profile for VoIP services.

```
zSH> CPE> VOIP> IP> add tripleplay
Service has been created
```

Creating a CPE VoIP subscriber profile and associate it with a VoIP server

To create a CPE VoIP subscriber profile on ONU POTS ports, use the **cpe voip add** command.

When creating a CPE VoIP subscriber profile:

- A VoIP server profile is required to associate the VoIP server information to the POTS port. There is no default VoIP server profile.
- A VoIP features profile and a VoIP media profile are also required when creating the CPE VoIP subscriber profile, if users do not specify these two profiles, then the default profiles are used.
- The POTS port ID specified in this command must match the port ID assigned during the creation of the subscriber facing MXK bridge and CPE connection.

- The dial-number, username, and password are required for SIP configuration.

To create a CPE VoIP subscriber profile on an ONU with the **cpe voip add** command:

- 1 Create SIP services on the “tripleplay” service template for POTS 1, and associate the “sip-test” VoIP server profile.

```
zSH> CPE> VOIP> add tripleplay/1 dial-number 1111
username 1111 password 1234 voip-server-profile
sip-test
Service has been created
```

- 2 Create SIP services on the “tripleplay” service template for POTS 2, and associate the “sip-test” VoIP server profile.

```
zSH> CPE> VOIP> add tripleplay/2 dial-number 1112
username 1112 password 1234 voip-server-profile
sip-test
Service has been created
```

Performing other Voice related configuration (Optional)

To perform the other voice related configuration, such as CPE IP common profile, CPE VoIP feature profile, CPE VoIP media profile, CPE SIP dial plans, CPE VoIP subscriber profile, refer to [Create VoIP on Dynamic OMCI on Uplink and Downlink Bridges on page 763](#).

View All Services Created on the Service Template

To view all services created on the service template, use the **srvtmpl show** command.

```
zSH> SRVTMPL> show tripleplay
CPE 0/0/3      tripleplay/gpononu
Service: DATA
GEM  UNI      UNI-VLAN/SLAN  VLAN/SLAN(COS,VID)  G-VLAN  Admin  Oper  Rg-Mode
-----
257  eth 1      0,102/----    0      up      B-Routed
257  eth 2      0,102/----    0      up      B-Routed
Service: IPTV
GEM  UNI      UNI VLAN/SLAN  OLT VLAN/SLAN  G-VLAN  Admin  Oper  Video Prof Rg-Mode
-----
258  eth 3      0,203/----    0      up      Bridged
Service: SIP
GEM  UNI      UNI VLAN/SLAN  OLT VLAN/SLAN  G-VLAN  Host IP      IP Srvr Prof
-----
259  pots      0,304/----    0
Port Admin Oper  Srvr Prof Feature Prof Media Prof DN      User Name
=====
1      up      1      1      1      1111      1111
2      up      1      1      1      1112      1112
```

Create AutoConfiguration Rules for the Service Template

AutoConfiguration rule profile is used to specify which service template will be applied on which ONU(s) when the ONU(s) coming online.

Users can use the **srvtmpl rule add** command to create AutoConfiguration rules.

Command:

```
srvtmpl rule add <profile-name>
    [ admin-state < up | down > ]
    [ priority < priority level > ]
    [ match-expression < 64 byte character string > ]
]

    [ service-template < service-template-name > ]
    [ target-uni < 32 byte character string > ]
    [ del-before-apply < true | false > ]
```

[Table 96](#) provides the description for command options in the **srvtmpl rule add** command.

Table 94: srvtmpl rule add Command Option Explanations

Command Option	Description
<i>profile-name</i>	Specifies service template rule profile description. A unique 36-char string.
admin-state <up down>	Up or Down. A rule will not participate in matching unless its admin state is Up. Possible values are up, down. Default value is up.
priority value	Sets the priority for this rule. When multiple rules at different priority levels are matched, the ones with the highest priority will be applied on the ONU. When multiple rules at the same priority level are matched, they are all applied on the ONU. Possible values are 1...N. Note that 1 is the lowest priority. Default value is 1.

Table 94: `srvtmpl rule add` Command Option Explanations

Command Option	Description
match-expression <i>value</i>	Defines logic expressions to be used in matching the ONU(s) this service template should be applied on. 256-char string. Values: model ONU model ID(s). If more than one model is specified , separated them with comma, e.g. model 2424, 2424a me ME profile name(s). If more than one model is specified, separated them with comma, e.g. me zhong-2424, zhong-2424-1 slot Slot number to apply, single or wildcard, e.g. slot 2, 11-14 olt OLT port to apply, single or wildcard, e.g. 2-4,8 Match-expression can be empty. That is a catch-all rule. If there are no higher priority rules, that rule is applied on any supported ONUs that come online. Examples: model 2424 slot 12 olt 2-5,8 model 2424 slot 2, 11-14
servie-template <i>name</i>	The service template that is applied on the ONU.
target-uni <i>value</i>	A 32-byte character field specifying which UNIs on the CPE should be applied to. Examples: eth 1-3,8 pots all Note: If this field is empty (default), then the entire source CPE is applied to the target without any altering of the UNI configurations.
del-before-apply <true false>	True or False. When it is set to true, old services are deleted before this service template is applied. Default is true.

To create a AutoConfiguration rule for the “tripleplay” service template:

- 1 Create a service template rule “tripleplayrule1”, which will apply “tripleplay” service template to all connected ONU 2426A.

```
zSH> srvtmpl rule add tripleplayrule1 match-expression "model 2426A" service-template tripleplay
```

Profile "tripleplayrule1" has been created with index 1

- 2 Show the settings of the service template rule profile.

```
zSH> srvtmpl rule show tripleplayrule1
```

```

Index      Profile Name      admin-state  Service Template  priority
del-before-apply
=====
1      tripleplayrule1      up           tripleplay        1
true
Match Expression : model 2426A
```

Target UNI :
1 entries found.

Turn on AutoDiscovery and AutoConfiguration in the MXK

Users can use **update cpe-cfg-global-settings 0** command to enable or disable the AutoDiscovery and AutoConfiguration. By default, they are disabled.

Enabling AutoDiscovery and AutoConfiguration in the MXK

Turn on AutoDiscovery and AutoConfiguration in the MXK:

```
zSH> update cpe-cfg-global-settings 0
cpe-cfg-global-settings 0
Please provide the following: [q]uit.
auto-assign: ---> {disabled}: enabled
auto-config: ---> {disabled}: enabled
params-string: -> {}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
```

Verify AutoDiscovery and AutoConfiguration on ONUs

Verifying AutoDiscovery and AutoConfiguration on ONUs

- 1 Physically connect ONUs to the OLT.

Note that this step is done in the customer site, usually after AutoDiscovery and AutoConfiguration enabled in the Central Office (CO).

In this test case, users connect two ONUs to the OLT, one is zNID 2426A, another one is zNID 2425A.

- 2 Verify that serial numbers, model IDs, and internal ME profiles have been automatically assigned and retrieved on ONUs.

The following command shows model ID 2425A, Serial Number ZNTS 0338ABDF, and internal ME profile zhone-2425 are assigned to the ONU ID 6/1/1.

```
zSH> onu show 6/1/1
```

ONU	Name	Enabled	Model #	Serial Number	OMCI files and profiles
1	1-6-1-1	Yes	2425A	ZNTS 0338ABDF	ME zhone-2425

Note : NULL Model String indicates not able to get model ID

The following command shows model ID 2426A, Serial Number ZNTS 033B1613, and internal ME profile zhone-2426 are assigned to the ONU ID 6/1/2.

```
zSH> onu show 6/1/2
```

ONU	Name	Enabled	Model #	Serial Number	OMCI files and profiles
2	1-6-1-2	Yes	2426A	ZNTS 033B1613 ME	zhone-2426

Note : NULL Model String indicates not able to get model ID

3 Verify that ONUs 6/1/1 and 6/1/2 are activated.

```
zSH> onu status 6/1
```

Processing list of 64

This command may take several minutes to complete.

Do you want to continue? (yes or no) [no] **yes**

slot 6 olt 1

ID	Onu	OperStatus	ConfigState	Download State	OLT Rx Power	ONT Rx Power	Distance (KM)	Gpon OnuStatus	AutoConfig State
1	1-6-1-1	Up	Active	NoUpgrade	-14.0 dBm	-13.3 dBm	0.1592	Active	Done
2	1-6-1-2	Up	Active	NoUpgrade	-13.0 dBm	-10.9 dBm	0.1607	Active	Done
3	1-6-1-3	(disabled)	-	-	-	-	-	-	-
4	1-6-1-4	(disabled)	-	-	-	-	-	-	-
5	1-6-1-5	(disabled)	-	-	-	-	-	-	-
6	1-6-1-6	(disabled)	-	-	-	-	-	-	-
7	1-6-1-7	(disabled)	-	-	-	-	-	-	-
8	1-6-1-8	(disabled)	-	-	-	-	-	-	-
9	1-6-1-9	(disabled)	-	-	-	-	-	-	-
10	1-6-1-10	(disabled)	-	-	-	-	-	-	-
11	1-6-1-11	(disabled)	-	-	-	-	-	-	-
12	1-6-1-12	(disabled)	-	-	-	-	-	-	-
13	1-6-1-13	(disabled)	-	-	-	-	-	-	-
14	1-6-1-14	(disabled)	-	-	-	-	-	-	-

4 Verify that only the ONUs that are specified in the service template rule are automatically configured.

This example shows ONU 6/1/1 with model ID 2425A is not configured, because the service template rule “tripleplayrule1” only allow ONUs with model ID 2426A to be auto-configured.

```
zSH> cpe show 6/1/1
```

No provisioning exists on CPE 1-6-1-1/gpononu

This example shows ONU 6/1/2 with model ID 2426A has been auto-configured with service template “tripleplay” according to the service template rule “tripleplayrule1”.

```
zSH> cpe show 6/1/2
```

CPE 6/1/2 1-6-1-2/gpononu

Service: DATA

GEM	UNI	UNI-VLAN/SLAN	VLAN/SLAN (COS,VID)	G-VLAN	Admin	Oper	Rg-Mode
257	eth 1		0,102/----	0	up	down	B-Routed
257	eth 2		0,102/----	0	up	down	B-Routed

Service: IPTV

GEM	UNI	UNI VLAN/SLAN	OLT VLAN/SLAN	G-VLAN	Admin	Oper	Video Prof	Rg-Mode
258	eth 3		0,203/----	0	up	down		Bridged

Service: SIP

GEM	UNI	UNI VLAN/SLAN	OLT VLAN/SLAN	G-VLAN	Host IP	IP Srvr Prof


```

259 pots 0,304/---- 0 1
Port Admin Oper Srvr Prof Feature Prof Media Prof DN User Name
=====
1 up down 1 1 1 1111 1111
2 up down 1 1 1 1112 1112

```

5 Verify the bridge configuration and CPE connections:

zSH> **bridge show**

```

Orig
Type VLAN/SLAN VLAN/SLAN Physical Bridge
St Table Data
-----
dwn Tagged 102 1/6/1/2/gpononu 1-6-1-257-gponport-102/
bridge UP D 3a:02:71:3b:16:18
dwn-vid Tagged 203 1/6/1/2/gpononu 1-6-1-258-gponport-203/
bridge UP
dwn Tagged 304 1/6/1/2/gpononu 1-6-1-259-gponport-304/
bridge UP D 00:02:71:3b:16:14
upl Tagged 203 1/a/4/0/eth ethernet4-203/bridge
DWN S VLAN 203 default
upl Tagged 304 1/a/5/0/eth ethernet5-304/bridge
DWN S VLAN 304 default
upl Tagged 102 1/a/8/0/eth ethernet8-102/bridge
DWN S VLAN 102 default
6 Bridge Interfaces displayed

```

zSH> **bridge show onu**

```

GEM ONU DSCP ONU UNI OLT OLT
ONU Port UNI to COS VLAN/SLAN VLAN/SLAN G-VLAN MVR Service Rg-Mode
OLT Bridge ST
-----
6/1/2 257 eth 1 ----- /---- Tagged 102 ---- data B-Routed
1-6-1-257-gponport-102/bridge DWN
6/1/2 257 eth 2 ----- /---- Tagged 102 ---- data B-Routed
1-6-1-257-gponport-102/bridge DWN
6/1/2 258 eth 3 ----- /---- Tagged 203 ---- video Bridged
1-6-1-258-gponport-203/bridge DWN
6/1/2 259 pots ----- /---- Tagged 304 ---- sip Bridged
1-6-1-259-gponport-304/bridge UP
3 Bridge Interfaces displayed
4 GPON ONU Connections displayed

```

6 Verify the assigned GEM port IDs and assigned GTP IDs:

zSH> **gpononu gemparts 6/1**

Processing list of 64

This command may take several minutes to complete.

Do you want to continue? (yes or no) [no] **yes**

Slot 6 olt 1

```

traf Fixed UBR Fixed CBR Assured Max Extra
Bandwidth Bandwidth Bandwidth Bandwidth Bandwidth

```

```

      ONU      GEM Port  Admin   prof   compn share Mbits/sec Mbits/sec Mbits/sec Mbits/sec Type      allocId
DBA
=====
=====
      1-6-1-2   1-6-1-257   Up     10240 False False    0.128      0    10.240    20.480 Nonassured    256
NSR
      1-6-1-259   Up       512 False False    0.128      0     0.384     0.512 Nonassured    258
NSR
      1-6-1-258   Up 11000000000 False False    0.128      0         0     1000 Nonassured    257
NSR
Total Available BW: 1123.290(Mb) , Total Available BW for Compensated CBR:
454.246(Mb)

```

Modify the Subscriber-Specific Settings

Users can modify the subscriber specific settings after the ONUs have been auto-discovered and auto-configured, such as VoIP Phone number, user name, password, WiFi SSID, encrypt-key, device-pin etc.

Modifying the parameters specific to the subscribers

- 1 Modify dial-number, username, password for ONU 6/1/2 POTs 1 and POTS 2:

```
zSH> cpe voip show all
```

```

      Port  Admin      Voip-Server  Voip Features  Voip Media
      CPE   Number State Rx Gain Tx Gain Profile Index Profile Index Profile Index
=====
      6/1/2    1    up      0      0          1          1          1
Dial Number   : 1111
Username      : 1111
Phone Follows Wan : disabled
      Port  Admin      Voip-Server  Voip Features  Voip Media
      CPE   Number State Rx Gain Tx Gain Profile Index Profile Index Profile Index
=====
      6/1/2    2    up      0      0          1          1          1
Dial Number   : 1112
Username      : 1112
Phone Follows Wan : disabled
2 services displayed

```

```
zSH> cpe voip modify 6/1/2/1 dial-number 7771234 username 7771234 password 12345
Service has been modified
```

```
zSH> cpe voip modify 6/1/2/2 dial-number 7771235 username 7771235 password 12345
Service has been modified
```

```
zSH> cpe voip show all
```

```

      Port  Admin      Voip-Server  Voip Features  Voip Media
      CPE   Number State Rx Gain Tx Gain Profile Index Profile Index Profile Index
=====
      6/1/2    1    up      0      0          1          1          1
Dial Number   : 7771234
Username      : 7771234
Phone Follows Wan : disabled
      Port  Admin      Voip-Server  Voip Features  Voip Media

```

CPE	Number	State	Rx Gain	Tx Gain	Profile Index	Profile Index	Profile Index
6/1/2	2	up	0	0	1	1	1

Dial Number : 7771235
 Username : 7771235
 Phone Follows Wan : disabled
 2 services displayed

2 Modify ssid, encrypt-key, and device-pin for ONU 6/1/2 WLAN port 1.

zSH> CPE> WLAN> **add 6/1/2/1 ssid zdev encrypt-key 1234567890 device-pin 1237**
 Service has been created

zSH> CPE> WLAN> **show 6/1/2/1**

CPE	WLAN	Admin	SSID	WLAN Com Prof	WLAN Com Adv Prof
6/1/2	1	up	zdev	1	1

1 services displayed

VoIP Phone with LLDP-MED Network Policy

LLDP

Link Layer Discovery Protocol (LLDP) got its start in an IETF Working Group (PTOPOMIB), which focused on a common framework or model to describe physical topology. This group published an informational MIB, RFC 2922, in September 2000. Cisco then worked with the IEEE to create 802.1AB, which was published in May 2005. This protocol provides standards-based discovery for vendor interoperability.

Examples of applications that use the information conveyed by discovery protocols include:

Network topology—A network management system (NMS) can accurately represent a map of the network topology.

Inventory—A management system can query a switch to learn about all the devices connected to that switch.

Emergency services—Location of a phone can be determined by the switch port to which it is connected.

VLAN configuration—The switch can tell the phone which VLAN to use for voice.

Power negotiation—The phone and the switch can negotiate the power that the phone can consume.

The LLDP packets include the following information about the zNID devices:

- Manufacturer
- Model Number
- Asset ID
- Serial Number
- MAC Address
- Port Description
- System Name
- System Location
- System Description
- HW Rev
- FW Rev
- Linux SW Rev
- PoE Power Capabilities
- Ethernet Link Speed and Duplex mode

- In addition, when an attached device responds with an LLDP packet that indicates support for the Media Endpoint Discovery (MED) extensions to LLDP, then the zNID will also send LLDP-MED packets.

LLDP-MED



Note: In the currently implementation, only Network Policies in LLDP-MED are Configurable.

It was recognized that additional capabilities were needed specific to VoIP, and this work was assumed by the Telecommunications Industry Association (TIA) TR-41.4 subcommittee. This committee developed the extension Media Endpoint Discovery to the Link Layer Discovery Protocol (LLDP-MED). LLDP-MED is specified to operate only between endpoint devices such as IP phones and network connectivity devices such as switches.

LLDP-MED uses Ethernet Multicast packets to enable communication between LLDP-enabled devices on a Bridged network.

Some LLDP-MED objects are configured on the ONT to define the content of the LLDP-MED frames sent out to the directly attached device. Some LLDP-MED objects are used to store information contained in LLDP-MED packets received from the directly attached device.

There is a 1:1 relationship between LAN ports on the ONT and LLDP-MED devices. Every physical Ethernet port on the ONT has an LLDP-MED interface instance in the database.

Configuring LLDP-MED Network Policies

LLDP-MED network policies can be created and configured from the **CPE>LLDP** menu. From this menu the users have access to the show, add, modify, delete, and find commands as below:

```
cpe lldp < add | delete | find | modify | show >
```

```
zSH> CPE> LLDP> help
```

```
cpe lldp show < < list-index | list-profile-name >[/<
entry-index >] > | < all >
```

```
Display single or multiple profiles
```

```
cpe lldp add < list-profile-name >
```

```
[ admin-state < up | down > ]
```

```
[ app-type < voice | voice-signal | guest-voice |
```

```
guest-voice-signal | softphone |
```

```
video-conference |
```

```
video-streaming | video-signal > ]
```

```
[ vlan-id < VLAN ID > ]
```

```
[ cos < COS > ]
```

```
[ dscp < DSCP > ]
```

This command creates a new profile. The `<list-profile-name>` must be supplied and must be unique for profile type. The profile index will be automatically generated

```
cpe lldp modify < list-index | list-profile-name >/< entry-index > [ arguments ]
```

Modify a profile. Must supply list index or name as well as entry index.

See "add" command for available [arguments]

```
cpe lldp delete < list-index | list-profile-name >/< entry-index >
```

Delete a profile. Must supply list index or name as well as entry index.

```
cpe lldp find < list-index | list-profile-name >
```

Find all cpe-eth-subscriber profiles that reference the cpe-lldp-med-policy Profile

The settings of the LLDP-MED network policy is listed in [Table 95](#).

Table 95: cpe lldp add Command Options Explanation

Command Option	Description
<i>list-profile-name</i>	The list name of the LLDP-MED network policy. Values: 36-char string
admin-state <up down>	Enable/disable the network policy. Default: up
app-type <voice voice-signal guest-voice-signal softphone video-conference video-streaming video-signal>	The media type that defines the primary function of the application for the policy advertised by an endpoint. Values: voice voicesignal guestvoice guestvoicesignal softphone videoconf streamingvideo videosignal
vlan-id <i>VLANID</i>	The VLAN ID of the application for the policy advertised by an endpoint. Values: 0-4095 Default: 0

Table 95: cpe lldp add Command Options Explanation

Command Option	Description
cos <i>COS</i>	The Class-Of-Service of the application for the policy advertised by an endpoint. Values: 0-7 Default: 0
dscp <i>DSCP</i>	The Differentiated Service Code Point of the application for the policy advertised by an endpoint. Values: 8-char string, 0-63 DSCP values or names

The following procedure shows how to configure an IP-phone with the LLDP-MED network policy in MXK.

- 1 Create a new LLDP-MED network policy for voice (IP-phone) service and add it to a specified list (if the list doesn't exist, a new list with the specified name is added to the system, and the profile index will be generated automatically). This command requires a unique list name for this profile type, and additionally accepts field specifications as [Table 95](#).

```
zSH> CPE> LLDP> add ipphone app-type voice vlan-id 8 cos 1
Profile "ipphone" has been created with index 2
Profile "ipphone"/1 has been created
```

```
zSH> CPE> LLDP> add ipphone app-type voicesignal vlan-id 8 cos 4
Profile "ipphone"/2 has been created
```

- 2 Show a LLDP-MED network policy list:

```
zSH> CPE> LLDP> show 2
  Index      Profile Name      admin-state  app-type  vlan-id  cos
dscp
=====
=====
2/1         ipphone              up           voice      8
1           0
  Index      Profile Name      admin-state  app-type  vlan-id  cos
dscp
=====
=====
2/2         .                    up           voicesignal 8
4           0
2 entries found.
```

- 3 If necessary, modify a LLDP-MED network policy list.

```
zSH> CPE> LLDP> modify 2/2 admin-state down
Profile has been modified.
```

- 4 Set an ME profile on the ONU:

```
zSH> onu set 4/1/1 meprof zhone-2648p
Onu 4/1/1 has been set
```

5 Create an uplink bridge and a GPON downlink bridge.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 8 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-8/bridge
Bridge-path added successfully
```

Note that the downlink bridge is created as for data service; users don't need to specify **voice** keyword for the IP phone with LLDP-MED network policy.

```
zSH> bridge add 1-4-1-1/gpononu gem 302 gtp 1 t1s vlan 8 tagged uni-vlan 8 eth 1
rg-bridged
Adding bridge on 1-4-1-1/gpononu
Created bridge-interface-record 1-4-1-302-gponport-8/bridge
Bridge-path added successfully
CPE Connection 1-4-1-302/gponport/1/1/8/0 has been created
```

6 Associate the LLDP-MED network policies with Ethernet UNI port 1 of the ONU:

```
zSH> cpe eth add 4/1/1/1 lldp-med-list ipphone
Service has been created
```

```
zSH> cpe eth show 4/1/1
```

CPE	Port Number	Admin	Rate	Duplex	Video Profile	Traf Mngt Profile	Alm St	Power Sev	Power Shed	Power Range	LLDP-MED List
4/1/1	1	up	auto	auto	0	0	Dis	Mj	En	Low	2

1 services displayed

7 If users want to find which ethernet UNI port is using the specific LLDP-MED list, use the **cpe lldp find** command.

```
zSH> CPE> LLDP> find ipphone
cpe-eth-subscriber 1
1 profiles displayed
```

8 Once the ONU is activated and an IP phone is connected to Ethernet UNI port 1, the ONU will notify the phone the VLAN and COS it should use. The IP phone then starts communicating with the VoIP switches on the VLAN.

Additional Features in Unified Service Provisioning with “bridge add” Command

- [VLAN translation on ONU on page 885](#)
- [DSCP to COS mapping on page 889](#)
- [Support UNI range in “bridge” command on page 891](#)
- [Support RG CoS in “bridge” command on page 896](#)
- [Create an RG-bridged connection without LAN members on page 897](#)
- [Create an RG connection without creating a VLAN in RG on page 898](#)

VLAN translation on ONU

For GPON line card, VLAN translation is performed on the ONU instead of on the MXK. VLAN translation on ONU is also referred to as CPE VLAN translation in this section.

In situations when devices in the core network expect unique identifiers for each subscriber, and because subscriber configurations on the MXK can include large numbers of subscribers with pre-configured VLAN IDs, the ONU supports VLAN translation from the subscriber to the MXK for VLANs sent to the core network.

When configuring bridges for VLAN translation, all network facing Ethernet ports on the MXK must be tagged, all bridges facing the subscriber’s ONU must be tagged, and all subscriber facing Ethernet UNI ports on the ONU must be tagged as well. Bridges that are untagged do not support translation. For VLAN translation to work, there must be a VLAN in the Ethernet packet when it arrives at the ONU from the subscriber facing Ethernet UNI port.

The range for translated VLAN IDs is 1-3897 for GPON line cards.

Possible bridging configuration behaviors for CPE VLAN translation

Possible bridging configuration behaviors for CPE VLAN translation:

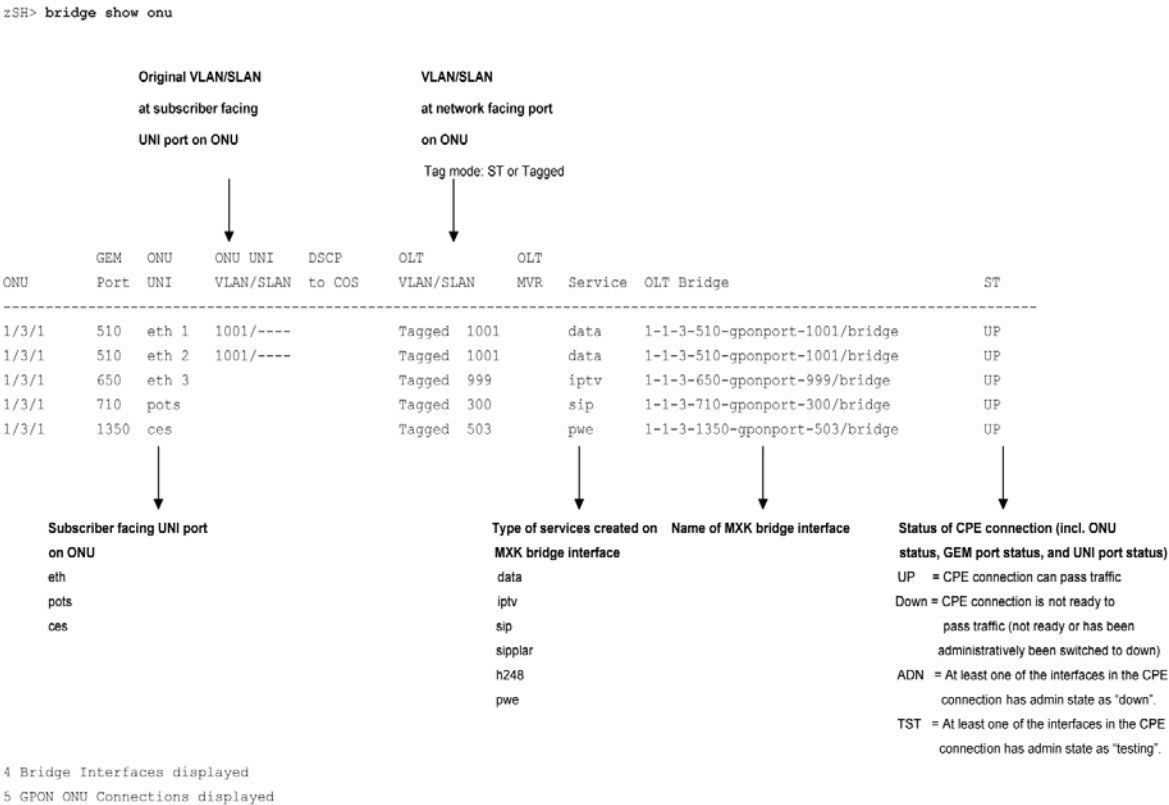
- the single-tagged Ethernet packet from subscriber facing ONU Ethernet UNI port, to subscriber facing single-tagged bridge, and to network facing single-tagged bridge with VLAN translation (*single-tagged to single-tagged*)

Refer to [CPE VLAN translation on uplink and downlink bridges on page 886](#).

bridge show onu command for CPE VLAN translation

The **bridge show onu** command displays both subscriber facing original VLAN IDs and the translated network facing VLAN IDs on ONU.

Figure 123: sample bridge show onu command display



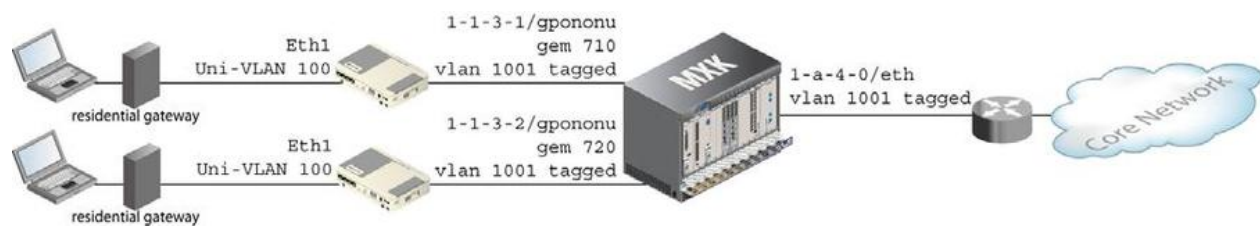
CPE VLAN translation on uplink and downlink bridges

This section describes configuring uplink/downlink bridges on the MXK and configuring CPE connections on the ONU for basic VLAN translation.

When configuring the ONU for VLAN translation, users *must* designate both the uplink bridge and the downlink bridge on the MXK as *tagged*, and the Ethernet UNI port on the ONU as *tagged* as well. And the translated VLAN ID must be same at both the uplink bridge and the downlink bridge on the MXK. This allows the original VLAN ID on the subscriber side of the ONU to pass down to the subscriber, and the translated VLAN ID on the network side of the ONU to pass to the MXK and the core network.

As shown in Figure 124, the VLAN ID 100 on subscriber facing Ethernet UNI port are translated on the ONU to VLAN ID 1001 for the subscriber facing downlink bridge and the network facing uplink bridge on the MXK.

Figure 124: Asymmetric bridges with CPE VLAN translation



Configuring single-tagged to single-tagged asymmetric bridges for CPE VLAN translation

- 1 Create a *tagged* uplink bridge with VLAN ID on the network facing Ethernet port.

```
zSH> bridge add 1-a-4-0/eth uplink vlan 1001 tagged
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-1001/bridge
Bridge-path added successfully
```

Verify the bridge.

```
zSH> bridge show
      Orig
Type VLAN/SLAN      VLAN/SLAN  Bridge                               St  Table Data
-----
upl                Tagged 1001 ethernet4-1001/bridge             UP S VLAN 1001 default
1 bridges displayed
```

- 2 Create *tagged* downlink bridges on the MXK with the translated VLAN ID, and create CPE connections with UNI-VLAN ID on subscriber facing Ethernet UNI ports.

This example translates uni-vlan 100 to vlan 1001.

```
zSH> bridge add 1-1-3-1/gpononu gem 710 gtp 1 downlink vlan 1001 tagged eth
1 uni-vlan 100
Adding bridge on 1-1-3-1/gpononu
Created bridge-interface-record 1-1-3-710-gponport-1001/bridge
```

```
zSH> bridge add 1-1-3-2/gpononu gem 720 gtp 1 downlink vlan 1001 tagged eth
1 uni-vlan 100
Adding bridge on 1-1-3-2/gpononu
Created bridge-interface-record 1-1-3-720-gponport-1001/bridge
```

Verify the downlink bridges. The **bridge show** command displays the VLAN ID the ONU translated.

```
zSH> bridge show vlan 1001
      Orig
Type VLAN/SLAN  VLAN/SLAN      Physical                               Bridge
St  Table Data
-----
dwn                Tagged 1001  1/1/3/2/gpononu                     1-1-3-720-gponport-1001/
bridge UP
dwn                Tagged 1001  1/1/3/1/gpononu                     1-1-3-710-gponport-1001/
bridge UP
upl                Tagged 1001  1/a/4/0/eth                         ethernet4-1001/bridge
UP S VLAN 1001 default
3 Bridge Interfaces displayed
```

Verify the CPE connections. The **bridge show onu** command displays the original VLAN ID of the ONU (i.e. under the column of ONU UNI VLAN/SLAN) and the VLAN ID the ONU translated (i.e. under the column of OLT VLAN/SLAN).

```
zSH> bridge show onu vlan 1001
```

ONU	GEM Port	ONU UNI	DSCP to COS	ONU UNI VLAN/SLAN	OLT VLAN/SLAN	OLT MVR	Service	OLT
Bridge				ST				
1/3/1	710	eth 1		100/----	Tagged 1001		data	
1-1-3-710-gponport-1001/bridge UP								
1/3/2	720	eth 1		100/----	Tagged 1001		data	
1-1-3-720-gponport-1001/bridge UP								
2 Bridge Interfaces displayed								
2 GPON ONU Connections displayed								

Deleting single-tagged to single-tagged asymmetric bridges with CPE VLAN translation

- 1 View the existing bridges and CPE connections.

```
zSH> bridge show vlan 1001
```

Orig	Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge
St	Table	Data			
dwn		Tagged 1001	1/1/3/2/gpononu		1-1-3-720-gponport-1001/
bridge UP					
dwn		Tagged 1001	1/1/3/1/gpononu		1-1-3-710-gponport-1001/
bridge UP					
upl		Tagged 1001	1/a/4/0/eth		ethernet4-1001/bridge
UP S VLAN 1001 default					
3 Bridge Interfaces displayed					

```
zSH> bridge show onu vlan 1001
```

ONU	GEM Port	ONU UNI	DSCP to COS	ONU UNI VLAN/SLAN	OLT VLAN/SLAN	OLT MVR	Service	OLT
Bridge				ST				
1/3/1	710	eth 1		100/----	Tagged 1001		data	
1-1-3-710-gponport-1001/bridge UP								
1/3/2	720	eth 1		100/----	Tagged 1001		data	
1-1-3-720-gponport-1001/bridge UP								
2 Bridge Interfaces displayed								
2 GPON ONU Connections displayed								

- 2 Delete the downlink bridges and stacked CPE connections. CPE VLAN ID translation uses the ethernet port ID and original VLAN ID in the **bridge delete** syntax.

```
zSH> bridge delete 1-1-3-710-gponport-1001/bridge eth 1 uni-vlan 100
CPE Connection 1-1-3-710/gponport/1/1/100/0 has been deleted
1-1-3-710-gponport-1001/bridge delete complete
```

```
zSH> bridge delete 1-1-3-720-gponport-1001/bridge eth 1 uni-vlan 100
CPE Connection 1-1-3-720/gponport/1/1/100/0 has been deleted
1-1-3-720-gponport-1001/bridge delete complete
```

- 3 Delete the uplink bridge. Uses the translated VLAN ID in the **bridge delete** syntax.

```
zSH> bridge delete ethernet4-1001/bridge vlan 1001
Bridge-path deleted successfully
ethernet4-1001/bridge delete complete
```

DSCP to COS mapping

Users can assign DSCP to CoS mapping for new CPE connections, usually on the subscriber facing Ethernet port (ingress).

Assigning DSCP to CoS mapping to CPE connection

Assign DSCP to COS mapping to the Dynamic OMCI ONUs when creating the CPE connection by using the **bridge add** command:

- 1 View the mapping values in the **dscp-to-cos** profile.

```
zSH> get dscp-to-cos 1
dscp-to-cos 1
dscp00map8021p: -> {0}
dscp01map8021p: -> {0}
dscp02map8021p: -> {0}
dscp03map8021p: -> {0}
dscp04map8021p: -> {0}
dscp05map8021p: -> {0}
dscp06map8021p: -> {0}
dscp07map8021p: -> {0}
dscp08map8021p: -> {1}
dscp09map8021p: -> {1}
dscp10map8021p: -> {1}
dscp11map8021p: -> {1}
dscp12map8021p: -> {1}
dscp13map8021p: -> {1}
dscp14map8021p: -> {1}
dscp15map8021p: -> {1}
dscp16map8021p: -> {2}
dscp17map8021p: -> {2}
dscp18map8021p: -> {2}
dscp19map8021p: -> {2}
dscp20map8021p: -> {2}
dscp21map8021p: -> {2}
dscp22map8021p: -> {2}
dscp23map8021p: -> {2}
```

```

dscp24map8021p: -> {3}
dscp25map8021p: -> {3}
dscp26map8021p: -> {3}
dscp27map8021p: -> {3}
dscp28map8021p: -> {3}
dscp29map8021p: -> {3}
dscp30map8021p: -> {3}
dscp31map8021p: -> {3}
dscp32map8021p: -> {4}
dscp33map8021p: -> {4}
dscp34map8021p: -> {4}
dscp35map8021p: -> {4}
dscp36map8021p: -> {4}
dscp37map8021p: -> {4}
dscp38map8021p: -> {4}
dscp39map8021p: -> {4}
dscp40map8021p: -> {5}
dscp41map8021p: -> {5}
dscp42map8021p: -> {5}
dscp43map8021p: -> {5}
dscp44map8021p: -> {5}
dscp45map8021p: -> {5}
dscp46map8021p: -> {5}
dscp47map8021p: -> {5}
dscp48map8021p: -> {6}
dscp49map8021p: -> {6}
dscp50map8021p: -> {6}
dscp51map8021p: -> {6}
dscp52map8021p: -> {6}
dscp53map8021p: -> {6}
dscp54map8021p: -> {6}
dscp55map8021p: -> {6}
dscp56map8021p: -> {7}
dscp57map8021p: -> {7}
dscp58map8021p: -> {7}
dscp59map8021p: -> {7}
dscp60map8021p: -> {7}
dscp61map8021p: -> {7}
dscp62map8021p: -> {7}
dscp63map8021p: -> {7}

```

2 Create a MXK bridge and its CPE connection, and assign DSCP to CoS mappings:

a Dynamic OMCI, untagged traffic

Use this format if users want to assign DSCP to CoS mapping in the untagged ingress traffic (i.e. without uni-vlan ID) on Ethernet Port.

```

zSH> bridge add 1-1-1-1/gpononu gem 301 gtp 1 dscp-to-cos 1 downlink vlan 100
tagged eth 1

```

Adding bridge on 1-1-1-1/gpononu

Created bridge-interface-record 1-1-1-301-gponport-100/bridge

CPE Connection 1-1-1-301/gponport/1/1/0/0 has been created

b Dynamic OMCI, tagged traffic

Use this format if users want to overwrite the CoS value with DSCP to CoS mapping in the tagged ingress traffic (i.e. with uni-vlan ID) on Ethernet Port 1.

```
zSH> bridge add 1-1-1-1/gpononu gem 301 gtp 1 dscp-to-cos 1 downlink vlan 100
tagged eth 1 uni-vlan 300
Adding bridge on 1-1-1-1/gpononu
Created bridge-interface-record 1-1-1-301-gponport-100/bridge
CPE Connection 1-1-1-301/gponport/1/1/300/0 has been created
```

c Residential Gateway, untagged traffic

Use this format if users want to assign DSCP to CoS mapping in the untagged ingress traffic (i.e. without uni-vlan ID) on Ethernet Port 1 in RG mode.

```
zSH> bridge add 1-1-1-1/gpononu gem 301 gtp 1 dscp-to-cos 1 downlink vlan 100
tagged eth 1 rg-bridged
Adding bridge on 1-1-1-1/gpononu
Created bridge-interface-record 1-1-1-301-gponport-100/bridge
CPE Connection 1-1-1-301/gponport/1/1/0/0 has been created
```

d Residential Gateway, tagged traffic

Use this format if users want to overwrite the CoS value with DSCP to CoS mapping in the tagged ingress traffic (i.e. with uni-vlan ID) on Ethernet Port 1 in RG mode.

```
zSH> bridge add 1-1-1-1/gpononu gem 301 gtp 1 dscp-to-cos 1 downlink vlan 100
tagged eth 1 uni-vlan 300 rg-bridged
Adding bridge on 1-1-1-1/gpononu
Created bridge-interface-record 1-1-1-301-gponport-100/bridge
CPE Connection 1-1-1-301/gponport/1/1/300/0 has been created
```

3 Verify DSCP to COS profile used in the CPE connection.

```
zSH> bridge show onu 1-1-1-1
```

ONU	GEM	ONU	DSCP	ONU UNI	OLT	OLT	G-VLAN	MVR	Service	Rg-Mode
OLT Bridge	Port	UNI	to COS	VLAN/SLAN	VLAN/SLAN					
1/1/1	301	eth 1	1	300/----	Tagged	100	----	----	data	Bridged
1-1-1-301-gponport-100/bridge					DWN					

1 Bridge Interfaces displayed
1 GPON ONU Connections displayed

Support UNI range in “bridge” command

In the **bridge add/delete** command for Unified Service Provisioning, the Ethernet UNI port ID and WLAN UNI port ID may be replaced with brackets containing numbers in comma-separated series (e.g **eth [1,3]**, **wlan [1,3]**), in

dash-separated ranges (e.g **eth [1-3]**, **wlan [1-3]**), and in wildcard (e.g **eth all**, **wlan all**).

The ethernet UNI port and WLAN port could be specified in the same **bridge add** command.

Note that WLAN port only could be specified in the **bridge add** command with the RG-mode.

The following sections provide examples for bridge add, show, and delete with UNI ranges:

- [Adding bridges with multiple interface ranges, page 892](#)
- [Deleting bridges with multiple interface ranges, page 895](#)

For the related information, refer to section [Bridge add commands with ranges of Slots, OLTs, GEM ports, and UNI ports, page 682](#).

Adding bridges with multiple interface ranges

- 1 Add bridges in RG mode with multiple interface ranges including UNI range.

```
zSH> bridge add 1-1-[1-3]-[301-304]/gponport gtp 1 downlink vlan 100 tagged eth
[1-2] rg-brouted
```

To Abort the operation enter Ctrl-C

Adding bridge on 1-1-1-301/gponport

Created bridge-interface-record 1-1-1-301-gponport-100/bridge

CPE Connection 1-1-1-301/gponport/1/1/0/0 has been created

CPE Connection 1-1-1-301/gponport/1/2/0/0 has been created

Adding bridge on 1-1-1-302/gponport

Created bridge-interface-record 1-1-1-302-gponport-100/bridge

CPE Connection 1-1-1-302/gponport/1/1/0/0 has been created

CPE Connection 1-1-1-302/gponport/1/2/0/0 has been created

Adding bridge on 1-1-1-303/gponport

Created bridge-interface-record 1-1-1-303-gponport-100/bridge

CPE Connection 1-1-1-303/gponport/1/1/0/0 has been created

CPE Connection 1-1-1-303/gponport/1/2/0/0 has been created

Adding bridge on 1-1-1-304/gponport

Created bridge-interface-record 1-1-1-304-gponport-100/bridge

CPE Connection 1-1-1-304/gponport/1/1/0/0 has been created

CPE Connection 1-1-1-304/gponport/1/2/0/0 has been created

Adding bridge on 1-1-2-301/gponport

Created bridge-interface-record 1-1-2-301-gponport-100/bridge

CPE Connection 1-1-2-301/gponport/1/1/0/0 has been created

CPE Connection 1-1-2-301/gponport/1/2/0/0 has been created

Adding bridge on 1-1-2-302/gponport

Created bridge-interface-record 1-1-2-302-gponport-100/bridge

CPE Connection 1-1-2-302/gponport/1/1/0/0 has been created

CPE Connection 1-1-2-302/gponport/1/2/0/0 has been created

Adding bridge on 1-1-2-303/gponport

Created bridge-interface-record 1-1-2-303-gponport-100/bridge

CPE Connection 1-1-2-303/gponport/1/1/0/0 has been created


```

CPE Connection 1-1-2-303/gponport/1/2/0/0 has been created
Adding bridge on 1-1-2-304/gponport
Created bridge-interface-record 1-1-2-304-gponport-100/bridge
CPE Connection 1-1-2-304/gponport/1/1/0/0 has been created
CPE Connection 1-1-2-304/gponport/1/2/0/0 has been created
Adding bridge on 1-1-3-301/gponport
Created bridge-interface-record 1-1-3-301-gponport-100/bridge
CPE Connection 1-1-3-301/gponport/1/1/0/0 has been created
CPE Connection 1-1-3-301/gponport/1/2/0/0 has been created
Adding bridge on 1-1-3-302/gponport
Created bridge-interface-record 1-1-3-302-gponport-100/bridge
CPE Connection 1-1-3-302/gponport/1/1/0/0 has been created
CPE Connection 1-1-3-302/gponport/1/2/0/0 has been created
Adding bridge on 1-1-3-303/gponport
Created bridge-interface-record 1-1-3-303-gponport-100/bridge
CPE Connection 1-1-3-303/gponport/1/1/0/0 has been created
CPE Connection 1-1-3-303/gponport/1/2/0/0 has been created
Adding bridge on 1-1-3-304/gponport
Created bridge-interface-record 1-1-3-304-gponport-100/bridge
CPE Connection 1-1-3-304/gponport/1/1/0/0 has been created
CPE Connection 1-1-3-304/gponport/1/2/0/0 has been created

```

2 View the created bridges.

```
zSH> bridge show vlan 100
```

Type	Orig VLAN/SLAN	St	Table	Data	VLAN/SLAN	Physical	Bridge
dwn					Tagged 100	1/1/1/1/gpononu	1-1-1-301-gponport-100/
bridge	UP						
dwn					Tagged 100	1/1/1/2/gpononu	1-1-1-302-gponport-100/
bridge	UP						
dwn					Tagged 100	1/1/1/3/gpononu	1-1-1-303-gponport-100/
bridge	DWN						
dwn					Tagged 100	1/1/1/4/gpononu	1-1-1-304-gponport-100/
bridge	UP						
dwn					Tagged 100	1/1/2/1/gpononu	1-1-2-301-gponport-100/
bridge	DWN						
dwn					Tagged 100	1/1/2/2/gpononu	1-1-2-302-gponport-100/
bridge	DWN						
dwn					Tagged 100	1/1/2/3/gpononu	1-1-2-303-gponport-100/
bridge	DWN						
dwn					Tagged 100	1/1/2/4/gpononu	1-1-2-304-gponport-100/
bridge	DWN						
dwn					Tagged 100	1/1/3/1/gpononu	1-1-3-301-gponport-100/
bridge	DWN						
dwn					Tagged 100	1/1/3/2/gpononu	1-1-3-302-gponport-100/
bridge	DWN						
dwn					Tagged 100	1/1/3/3/gpononu	1-1-3-303-gponport-100/
bridge	DWN						
dwn					Tagged 100	1/1/3/4/gpononu	1-1-3-304-gponport-100/
bridge	DWN						

12 Bridge Interfaces displayed

3 View the created CPE connections.

zSH> **bridge show onu vlan 100**

ONU	GEM Port	ONU UNI	DSCP to COS	ONU UNI VLAN/SLAN	OLT VLAN/SLAN ST	OLT G-VLAN	OLT MVR	Service	Rg-Mode
ONU OLT Bridge									
1/3/1	301	eth 1	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-3-301	gponport-100/bridge				DWN				
1/3/1	301	eth 2	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-3-301	gponport-100/bridge				DWN				
1/2/3	303	eth 1	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-2-303	gponport-100/bridge				DWN				
1/2/3	303	eth 2	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-2-303	gponport-100/bridge				DWN				
1/3/4	304	eth 1	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-3-304	gponport-100/bridge				DWN				
1/3/4	304	eth 2	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-3-304	gponport-100/bridge				DWN				
1/1/2	302	eth 1	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-1-302	gponport-100/bridge				DWN				
1/1/2	302	eth 2	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-1-302	gponport-100/bridge				DWN				
1/2/2	302	eth 1	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-2-302	gponport-100/bridge				DWN				
1/2/2	302	eth 2	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-2-302	gponport-100/bridge				DWN				
1/3/2	302	eth 1	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-3-302	gponport-100/bridge				DWN				
1/3/2	302	eth 2	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-3-302	gponport-100/bridge				DWN				
1/3/3	303	eth 1	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-3-303	gponport-100/bridge				DWN				
1/3/3	303	eth 2	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-3-303	gponport-100/bridge				DWN				
1/1/4	304	eth 1	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-1-304	gponport-100/bridge				UP				
1/1/4	304	eth 2	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-1-304	gponport-100/bridge				UP				
1/2/1	301	eth 1	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-2-301	gponport-100/bridge				DWN				
1/2/1	301	eth 2	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-2-301	gponport-100/bridge				DWN				
1/2/4	304	eth 1	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-2-304	gponport-100/bridge				DWN				
1/2/4	304	eth 2	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-2-304	gponport-100/bridge				DWN				
1/1/3	303	eth 1	-----	----/----	Tagged 100	----	----	data	B-Routed
1-1-1-303	gponport-100/bridge				DWN				

```

1/1/3      303 eth 2 ----- /---- Tagged 100 ---- data B-Routed
1-1-1-303-gponport-100/bridge DWN
12 Bridge Interfaces displayed
24 GPON ONU Connections displayed

```

4 View CPE.

```
zSH> cpe show 1/1/1
```

```
CPE 1/1/1
```

```
Service: DATA
```

GEM	UNI	UNI-VLAN/SLAN	VLAN/SLAN(COS,VID)	G-VLAN	Admin	Oper	Rg-Mode
301	eth 1		0,100/----	0	up		B-Routed
301	eth 2		0,100/----	0	up		B-Routed

Deleting bridges with multiple interface ranges

Delete bridges with multiple interface ranges including UNI range.

Note that if users have more than one CPE connection associated with those bridges users want to delete, users can either delete the CPE connection profiles separately or use the “all” command line argument.

1 Delete all CPE connection associated with Ethernet 1:

```
zSH> bridge delete 1-1-[1-3]-[301-304]/gponport eth 1
```

```
To Abort the operation enter Ctrl-C
```

```

CPE Connection 1-1-1-301/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-1-302/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-1-303/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-1-304/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-2-301/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-2-302/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-2-303/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-2-304/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-3-301/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-3-302/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-3-303/gponport/1/1/0/0 has been deleted
CPE Connection 1-1-3-304/gponport/1/1/0/0 has been deleted
0 bridge interfaces deleted out of 12 found

```

2 Delete bridge interfaces and all associated CPE connections:

```
zSH> bridge delete 1-1-[1-3]-[301-304]/gponport all
```

```
To Abort the operation enter Ctrl-C
```

```

CPE Connection 1-1-1-301/gponport/1/2/0/0 has been deleted
1-1-1-301/gponport delete complete
CPE Connection 1-1-1-302/gponport/1/2/0/0 has been deleted
1-1-1-302/gponport delete complete
CPE Connection 1-1-1-303/gponport/1/2/0/0 has been deleted
1-1-1-303/gponport delete complete
CPE Connection 1-1-1-304/gponport/1/2/0/0 has been deleted
1-1-1-304/gponport delete complete
CPE Connection 1-1-2-301/gponport/1/2/0/0 has been deleted
1-1-2-301/gponport delete complete

```

```

CPE Connection 1-1-2-302/gponport/1/2/0/0 has been deleted
1-1-2-302/gponport delete complete
CPE Connection 1-1-2-303/gponport/1/2/0/0 has been deleted
1-1-2-303/gponport delete complete
CPE Connection 1-1-2-304/gponport/1/2/0/0 has been deleted
1-1-2-304/gponport delete complete
CPE Connection 1-1-3-301/gponport/1/2/0/0 has been deleted
1-1-3-301/gponport delete complete
CPE Connection 1-1-3-302/gponport/1/2/0/0 has been deleted
1-1-3-302/gponport delete complete
CPE Connection 1-1-3-303/gponport/1/2/0/0 has been deleted
1-1-3-303/gponport delete complete
CPE Connection 1-1-3-304/gponport/1/2/0/0 has been deleted
1-1-3-304/gponport delete complete
12 bridge interfaces deleted out of 12 found

```

Support RG CoS in “bridge” command

The MXK supports setting CoS values in RG Ethernet VLAN headers for bridged packets. By specifying different CoS values for the services, a priority scheduler can drop the low priority traffic first if traffic exceeds the bandwidth.

The **cos** keyword in the **bridge add** command with **rgMode** option specifies the value loaded into the COS field of the VLAN header when an untagged packet received on the interface of a RG-provisioned zNID is tagged (VLAN ID inserted) for bridging. CoS values range from 0 — 7, with the lowest priority being 0 (default value) and the highest priority 7.

This example adds ONU 1/1/1 Ethernet interface 1 with a RG COS value of 5 in VLAN 200 for Data service:

```

zSH> bridge add 1-1-1-1 gem 301 gtp 1 downlink vlan 200 cos 5 tagged eth 1 rg-bridged
Adding bridge on 1-1-1-1
Created bridge-interface-record 1-1-1-301-gponport-200/bridge
CPE Connection 1-1-1-301/gponport/1/1/0/0 has been created

```

This example adds ONU 1/1/1 Ethernet interface 2 with a RG COS value of 6 in VLAN 300 for Video service:

```

zSH> bridge add 1-1-1-1 gem 401 gtp 1 downlink vlan 300 cos 6 tagged eth 2 rg-bridged
video 0/4
Adding bridge on 1-1-1-1
Created bridge-interface-record 1-1-1-401-gponport-300/bridge
CPE Connection 1-1-1-401/gponport/12/2/0/0 has been created

```

To view the CoS value in the MXK CLI, use the **cpe show** command, VLAN/SLAN (COS,VID) column. The displaying format of the VLAN/SLAN (COS, VID) column is VLAN CoS, VLAN ID/ SLAN CoS, SLAN ID. As shown in the following example, for Data service, VLAN CoS is 5 and VLAN ID is 200; for Video service, VLAN CoS is 6 and VLAN ID is 300.

```
zSH> cpe show 1/1/1
```

```
CPE 1/1/1
```

```
Service: DATA
```

GEM	UNI	UNI-VLAN/SLAN	VLAN/SLAN (COS,VID)	G-VLAN	Admin	Oper	Rg-Mode
301	eth 1		5,200/----	0	up		Bridged

```
Service: IPTV
```

GEM	UNI	UNI-VLAN/SLAN	VLAN/SLAN (COS,VID)	G-VLAN	Admin	Oper	Video Prof	Rg-Mode
401	eth 2		6,300/----	0	up		Bridged	

Create an RG-brridged connection without LAN members

Create an rg-brridged VLAN that is mapped to the ONU VEIP (Virtual Ethernet Interface Point) and does not have any LAN ports (i.e. UNI ports) as members.

The service type keyword **mgmt** in the **bridge add** command indicates this rg-brridged connection does not need to specify any UNI ports.

```
zSH> bridge add 1-1-1-1/gpononu gem 401 gtp 1 downlink vlan 102 tagged mgmt rg-brridged
```

```
zSH> bridge show onu 1-1-1-1
```

ONU	GEM	ONU	DSCP	ONU UNI	OLT	OLT	G-VLAN	MVR	Service	Rg-Mode	OLT
Bridge	Port	UNI	to COS	VLAN/SLAN	VLAN/SLAN						
1/1/1	401	mgmt	-----	----/----	Tagged	102	----	----	mgmt	Bridged	
1-1-1-401-gponport-102/bridge					DWN						

IP interface is not configured by default. It can be configured through WAN modification.

```
zSH> cpe rg wan show 1/1/1
```

CPE	VLAN/SLAN	RG Mode	IP Address	Auth	Retry Interval	Ip-Com Profile	Port-Fwd List	Port-Fwd Profile
G-VLAN								
1/1/1	104/----	Bridged	IP Unconfigured	--	--	0	0	--

```
Pppoe User Id: --
```

Associate IP Common Profile 1 with the WAN interface:

```
zSH> CPE> RG> WAN> modify 1/1/1 vlan 104 ip-com-profile 1
```

```
Service has been modified
```

As Host IP Option in IP Common Profile 1 is set to DHCP, now the WAN IP interface is configured for DHCP.

```
zSH> CPE> RG> WAN> show 1/1/1
```

CPE	VLAN/SLAN	RG Mode	IP Address	Retry Auth	Ip-Com Interval	Port-Fwd Profile List	Profile	G-VLAN
1/1/1	104/----	Bridged	dhcp	--	--	1	0	--

Pppoe User Id: --

Create an RG connection without creating a VLAN in RG

Create an RG connection between a GEM port and the ONU VEIP with OMCI, but without configure anything in RG (i.e. between the VEIP to an UNI port). This kind of connections can be used in the cases that RG configuration are done through TR-69 (it does not require the MXK to establish TR-69 management path), or third-party ONUs that do not support Zhone SNMP, etc.

The service type keyword **rg** in the **bridge add** command indicates an RG connection is created without creating a VLAN in RG.

```
zSH> bridge add 1-1-1-1 gem 601 gtp 1 downlink vlan 600 tagged rg
```

```
zSH> bridge show onu 1-1-1-1
```

ONU	GEM Port	ONU UNI	DSCP to COS	ONU UNI VLAN/SLAN	OLT VLAN/SLAN	G-VLAN	MVR	Service	Rg-Mode	OLT Bridge	ST
1/1/1	601	rg	-----	600/----	Tagged 600	----	----	rg	-----	1-1-1-601-gponport-600/bridge	UP

Post Configuration in USP



Note: Post configuration requires a post configuration script from Zhone. Please contact Zhone GSS to obtain a post configuration script suited to your needs.

After the CPE has been configured with Unified Service Provision, if users want to configure some extra features on this CPE, and those features do not have profile support, a post configuration script can be used to post-configure the CPE. The post configuration scripts are created by Zhone as needed without upgrading the MXK software.

Appending post script file to the ME Profile

To append supplemental post configuration script to the internal “Zhone-” ME profile, use the following procedure:

- 1 Load the post configuration script file into the MXK file system.
- 2 Append the post configuration script file into the internal ME profile.

```
zSH> onu profile append me zhone-9480 pw5.txt
```

- 3 Set the internal ME profile and its appendix for the ONT.

```
zSH> onu set 1/1/3 meprof zhone-9480
```

- 4 Configure the services on the ONT.

```
zSH> bridge add 1-1-1-1/gpononu gem 901 gtp 2 t1s vlan
60 tagged pwe rg-bridged
```

- 5 Must use the **onu apply** command to issue the OMCI configuration command in the ME profile.

```
zSH> onu apply 1/1/3
```



Note: If the post configuration script file contains variables, follow the procedure to add Generic and Specific profiles as needed.

Removing post configuration script file from the ME profile

To remove post configuration script from the internal “Zhone-” ME profile, use the **onu profile revert me** command.

```
zSH> onu profile revert me zhone-9480
```

ONU Software Upgrades

This section describes how to upgrade ONU software via OMCI method or TFTP method.

ONU Software Upgrades via OMCI

This section describes the ONU SW upgrade via OMCI.



Note: If the CPE model and its software version supports both TFTP and OMCI download method, MXK uses TFTP when it is possible.

- [Manual upgrade on an ONU, page 900](#)
- [Auto upgrade on an ONU, page 904](#)
- [View the ONU upgrade status, page 907](#)

Manual upgrade on an ONU

The OMCI image upgrade feature enables MXK user to manually upgrade firmware on an ONU immediately. Before downloading the image file to the ONU, make sure the image file exists in the MXK first. The OMCI standard defines two managed entities for firmware image, which are referred to as partitions 0 and partition 1.

These actions are supported by the OMCI image manual upgrade feature:

- Download
Download an image file from OLT to ONU.
- Activate
Reboot using an image in the specified partition.
- Commit
Set the image in the specified partition as the boot default.
- Abort
Terminate the queued download.

According to the OMCI standard, download is only allowed if the partition is neither active nor committed. Similarly, activate and commit are only allowed if the partition is valid.

After successfully downloading an image file, the ONU automatically checks whether the image file is valid. Only valid file can be activated.

Upgrade the firmware image on an ONU from the MXK with the **gpononu image** command.

```
gpononu image slot/olt/onu download filename | activate | commit |  
download-activate filename | download-activate-commit filename | abort |  
show [part partition#]
```


Table 96 provides the description for command options in the **gpononu image** command.

Table 96: Gpononu image Command Option Explanations

Command Option	Description
download <i>filename</i> [part <i>partition#</i>]	Download an image file to the ONU from the OLT. Part partition number is optional. An image file will be downloaded to either an inactive partition or an uncommitted partition. After downloading, the ONU validates the file.
activate [part <i>partition#</i>]	Bootup a valid file in the inactive partition immediately in the ONU. Part partition number is optional. Only one partition at a time can be active.
commit [part <i>partition#</i>]	Specify a default file to bootup the next time this ONU is powered up. Part partition number is optional. It will commit the file in the uncommitted partition. Only one partition at a time can be committed.
download-activate <i>filename</i> [part <i>partition#</i>]	Perform the download action, and then if the file passes the validation check, perform the activate action. Part partition number is optional.
download-activate-commit <i>filename</i> [part <i>partition#</i>]	Perform the download and activate actions, and then if the ONU ranges, perform the commit action. If ranging doesn't occur within a timeout period, return error. Part partition number is optional.
abort	Terminate the queued download.
show	Show the settings for the files downloaded. Users can view the file version, the validation status, the activation status, and the commitment status for each partition. It also provide download status, ONU model ID, Upgrade start time, will be activated or not, will be committed or not, and upgrade type.
part <i>partition#</i>	Users can have two image files stored in the ONU. One in partition 0, one in partition 1.

The following examples describe some common configurations during GPON ONU image upgrading:

Download image file *2510/2510.img* from MXK OLT to ONU 11/4/24, and activate-commit it at once with the **gpononu image slot/olt/onu download-active-commit** command.

Note that when downloading the image file to the ONU, the ONU image file must exists in the MXK flash. Not specifying the directory in the fileName field indicates the file is stored under the root directory, */card1*. And this example didn't specify the partition number, so the file will be downloaded to the current standby partition, partition 0.

```
zSH> gpononu image 11/4/24 download-activate-commit 2510/
2510.img
Image "/card1/2510/2510.img" downloading to onu 11/4/24.
```

View the upgrade status on this ONU with the **gpononu image slot/olt/onu show** command.

This example shows the image download has been requested, and has been queued by the system for download. The download status is Queued.

```
zSH> gpononu image 11/4/24 show
```

	Partition 0	Partition 1
Version:	R2.0.8.16	R2.0.8.17
isCommitted:	False	True
isActive:	False	True
isValid:	True	True
Download status:	Queued	
Onu model id:	2510	
Upgrade start time:	OCT 01 23:15:32 2009	
Will be activated:	True	
Will be committed:	True	
Upgrade type:	Manual	

This example shows the image download is in progress or validating the downloaded image is in progress. The download status is Downloading, and the isValid status in Partition 0 is False.

```
zSH> gpononu image 11/4/24 show
```

	Partition 0	Partition 1
Version:	SOFTWAREIMAGE0	R2.0.8.17
isCommitted:	False	True
isActive:	False	True
isValid:	False	True
Download status:	Downloading 10% complete	
Onu model id:	2510	
Upgrade start time:	OCT 01 23:15:32 2009	
Will be activated:	True	
Will be committed:	True	
Upgrade type:	Manual	

This example shows the image file has been downloaded to the ONU and passed validation, but not activated yet. The download status is Downloaded, and isValid status in Partition 0 is True.

```
zSH> gpononu image 11/4/24 show
```

	Partition 0	Partition 1
Version:	R2.0.8.18	R2.0.8.17
isCommitted:	False	True
isActive:	False	True
isValid:	True	True
Download status:	Downloaded	
Onu model id:	2510	
Upgrade start time:	OCT 01 23:15:32 2009	
Will be activated:	True	
Will be committed:	True	
Upgrade type:	Manual	

This example shows the image file has been activated. The isActive status is True.

```
zSH> gpononu image 11/4/24 show
```

	Partition 0	Partition 1
Version:	R2.0.8.18	R2.0.8.17
isCommitted:	False	True
isActive:	True	False
isValid:	True	True
Download status:	Downloaded	
Onu model id:	2510	
Upgrade start time:	OCT 01 23:15:32 2009	
Will be activated:	False	
Will be committed:	True	
Upgrade type:	Manual	

This example shows the whole downloading, activating, and committing the image file to the ONU is successfully completed. The isCommitted status is True, and the Download status is Complete.

```
zSH> gpononu image 11/4/24 show
```

	Partition 0	Partition 1
Version:	R2.0.8.18	R2.0.8.17
isCommitted:	True	False
isActive:	True	False
isValid:	True	True
Download status:	Complete	
Onu model id:	2510	
Upgrade start time:	OCT 01 23:15:32 2009	
Will be activated:	False	
Will be committed:	False	
Upgrade type:	Manual	

The possible values of the Download status are:

- None: ONU is not running, or not OMCI-provisioned.
- No Upgrade: ONU is running, but hasn't been upgraded yet.
- Queued: An Image download has been requested, and has been queued by the system for download.
- Downloading: Downloading the image to the ONU is in progress or validating the downloaded image is in progress.
- Downloaded: The software has been downloaded to the ONU but not activated yet. One possible reason is the ONU is rebooting.
- Complete: Successfully completed downloading, activating, and committing the software to the ONU.
- Error: Failed to upgrade due to some errors.
- Aborted: The ONU queued to be upgraded was aborted by request.

- FileErr: The software file to be upgraded does not exist, or has errors

Auto upgrade on an ONU

The OMCI ONU auto upgrade feature allows MXK users to automatically upgrade ONUs which are installed with the outdated software images when they are ranging.

If the ONU auto-upgrade is enabled, when an ONU ranges, MXK searches a auto-upgrade template for this ONU model. The auto-upgrade template is pre-defined by the user for each ONU model. It contains the auto-upgrade enable status, model ID, allowed software version, and the image file to be downloaded. If the matching template is found for the specific ONU model, the MXK compares the ONU software version with the allowed software version defined in the template. If they are same, then the auto-upgrade is interrupted, otherwise the MXK automatically upgrades the ONU.

The actions to automatically upgrade an ONU software through OMCI are Download->Activate->Commit.

The download is always performed on the standby partition. If the download is successful, then the standby partition is made the active and then the image is committed to the partition. After the image is committed, the auto-upgrade is finished.

Auto upgrading an ONU

1 Create an auto-upgrade template for an ONU model.

a Create an auto-upgrade template:

Note that when creating the template, the image file must already exist in the flash, otherwise an error message displays. Not specifying the directory in the *fileName* field indicates the file is stored under the root directory, */card1*.

```
zSH> new remote-sw-upgrade-profile 4
remote-sw-upgrade-profile 4
Please provide the following: [q]uit.
enabled: ---> {true}:
model: -----> {}: 2510
swVersion: -> (): R2.0.8.18
filename: --> (): zh.sip.cimg.2.0.8.18
.....
Save new record? [s]ave, [c]hange or [q]uit:s
```

b Verify the template is created:

```
zSH> list remote-sw-upgrade-profile
remote-sw-upgrade-profile 4
1 entry found.
```

2 Verify ONU auto-upgrade is enabled:

To perform ONU auto upgrade, both the ONU model auto-upgrade status and ONU ID auto-upgrade status should be enabled. By default, those two places are enabled.

- a** Verify the auto-upgrade status on an ONU model is enabled in the auto-upgrade template:

```
zSH> update remote-sw-upgrade-profile 4
remote-sw-upgrade-profile 4
Please provide the following: [q]uit.
enabled: ---> {true}:
model: -----> {2510}:
swVersion: -> (R2.0.8.18):
filename: --> (zh.sip.cimg.2.0.8.18):
.....
Save new record? [s]ave, [c]hange or [q]uit:q
```

- b** Verify the auto-upgrade status on an ONU ID is enabled in the gpon-olt-onu-config profile:

```
zSH> get gpon-olt-onu-config 1-15-1-1/gpononu
gpon-olt-onu-config 1-15-1-1/gpononu
serial-no-vendor-id: -----> {ZNTS}
serial-no-vendor-specific: -----> {0}
password: -----> {}
auto-learn: -----> {enabled}
power-level: -----> {0}
us-ber-interval: -----> {5000}
ds-ber-interval: -----> {5000}
onu-added: -----> {false}
omci-file-name: -----> {}
ONU-Managed-Entity-Profile-name: -----> {}
ONU-Generic-Assignments-Profile-name: -> {}
physical-traps: -----> {disabled}
ont-traps: -----> {disabled}
line-status-traps: -----> {disabled}
auto-upgrade: -----> {enabled}
serial-no-vendor-specific-fsan: -----> {0}
use-reg-id: -----> {disabled}
us-rx-power-monitoring-mode: -----> {monitoronly}
us-rx-power-high-threshold: -----> {-10}
us-rx-power-low-threshold: -----> {-30}
dba-status-reporting: -----> {disabled}
```

Or users can view it with the **gpononu auto-upgrade show slot [/olt/[onu]] | all** command:

```
zSH> gpononu auto-upgrade show 15
Processing list of 512
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
Slot 15 olt 1
ONU      Name      Auto-upgrade
=== =====
```

```

1          1-15-1-1      enabled
2          1-15-1-2      enabled
3          1-15-1-3      enabled
4          1-15-1-4      enabled
5          1-15-1-5      enabled

```

<SPACE> for next page, <CR> for next line, A for all, Q to quit

- c** In case the auto-upgrade status on this ONU ID is disabled, it can be enabled by the **gpononu auto-upgrade enable slot [/olt[/onu]] | all** command:

```

zSH> gpononu auto-upgrade enable 15/1
This command may affect many OLTs and ONUs.
Do you want to continue? (yes or no) [no] yes

processing 1-15-1-1/gpononu ...
processing 1-15-1-2/gpononu ...
processing 1-15-1-3/gpononu ...
processing 1-15-1-4/gpononu ...
processing 1-15-1-5/gpononu ...
...
Operation successful.

```

3 Disable ONU auto-upgrade.

Disabling auto-upgrade in the auto-upgrade template or on the individual ONU causes the MXK to abort the queued download.

- a** Disable auto-upgrade for an ONU model in the auto-upgrade template:

```

zSH> update remote-sw-upgrade-profile 4
remote-sw-upgrade-profile 4
Please provide the following: [q]uit.
enabled: ---> {true}: false
model: -----> {2510}:
swVersion: -> (R2.0.8.18):
filename: --> (zh.sip.cimg.2.0.8.18):
.....
Save new record?[s]ave, [c]hange or [q]uit:s

```

- b** Disable auto-upgrade on an ONU with the **gpononu auto-upgrade disable slot [/olt[/onu]] | all** command:

```

zSH> gpononu auto-upgrade disable 15/1
This command may affect many OLTs and ONUs.
Do you want to continue? (yes or no) [no] yes

processing 1-15-1-1/gpononu ...
processing 1-15-1-2/gpononu ...
processing 1-15-1-3/gpononu ...
processing 1-15-1-4/gpononu ...
processing 1-15-1-5/gpononu ...
...

```

Operation successful.

View the ONU upgrade status

- The ONU upgrade status for both auto-upgrade and manual upgrade per ONU can be viewed with the **gpononu upgrade show** command.

This command shows the ONU upgrade state (note that it is same as the Download status of the **gpononu image** command), ONU model ID, the date-time when last upgrade was started, active state, commit state, type of upgrade (Manl or Auto), and which partition is used.

- The detail partition information on each ONU can be viewed with the **gpononu image slot/olt/onu show** command.

Viewing ONU upgrade status

Here are the examples for each show command:

- View the upgrade status on an ONU:

```
zSH> gpononu upgrade show 11/4/2
```

ONU	State	Model	Start Time	Act	Cmt	Typ	Part
11/4/2	Complete	2510	OCT 01 19:37:39 2009	F	F	Auto	1

The possible values of the State field are:

- None: No download requested since slot boot.
- Noupgrade: ONU is running, but hasn't been upgraded yet.
- Queued: An Image download has been requested, and has been queued by the system for download.
- Downloading: Downloading the image to the ONU is in progress or validating the downloaded image is in progress.
- Downloaded: The software has been downloaded to the ONU but not activated yet. One probable reason is the ONU is rebooting.
- Complete: Successfully completed downloading, activating, and committing the software to the ONU.
- Error: Failed to upgrade due to some errors.
- Aborted: The ONU queued to be upgraded was aborted.
- FileErr: The software file to be upgraded does not exist, or has errors

- View upgrade status of the ONUs that have the same upgrade state:

```
zSH> gpononu upgrade show complete
```

ONU	State	Model	Start Time	Act	Cmt	Typ	Part
11/4/1	Complete	2510	OCT 01 22:28:40 2009	F	F	Manl	1
11/4/2	Complete	2510	OCT 01 19:37:39 2009	F	F	Auto	1

3 View the upgrade status and the partition information on an ONU:

```

zSH> gpononu image 11/4/2 show

```

	Partition 0	Partition 1
Version:	R2.0.8.16	R2.0.8.17
isCommitted:	False	True
isActive:	False	True
isValid:	True	True
Download status:	Complete	
Onu model id:	2510	
Upgrade start time:	OCT 01 19:37:39 2009	
Will be activated:	False	
Will be committed:	False	
Upgrade type:	Auto	

4 View all the ONU upgrade status on an OLT:

```

zSH> gpononu upgrade show 11/4

```

ONU	State	Model	Start Time	Act	Cmt	Typ	P
11/4/11	NoUpgrade	2510					
11/4/12	NoUpgrade	2510					
11/4/13	NoUpgrade	2510					
11/4/14	NoUpgrade	2510					
11/4/15	NoUpgrade	2510					
11/4/16	NoUpgrade	2510					
11/4/17	NoUpgrade	2510					
11/4/18	NoUpgrade	2510					
11/4/19	NoUpgrade	2510					
11/4/20	NoUpgrade	2510					
11/4/21	NoUpgrade	2510					
11/4/22	NoUpgrade	2510					
11/4/23	NoUpgrade	2510					
11/4/25	NoUpgrade	2510					
11/4/26	NoUpgrade	2510					
11/4/27	NoUpgrade	2510					
11/4/28	NoUpgrade	2510					
11/4/29	NoUpgrade	2510					
11/4/30	NoUpgrade	2510					
11/4/31	NoUpgrade	2510					
11/4/32	NoUpgrade	2510					
11/4/1	Complete	2510	OCT 01 22:28:40 2009	F	F	Manl	1
11/4/2	Complete	2510	OCT 01 19:37:39 2009	F	F	Auto	1
11/4/3	Aborted	2510	OCT 01 19:41:37 2009	F	F	Auto	1
11/4/4	Aborted	2510	OCT 01 19:41:38 2009	F	F	Auto	1
11/4/5	Aborted	2510	OCT 01 19:41:42 2009	F	F	Auto	1
11/4/6	Aborted	2510	OCT 01 19:41:46 2009	F	F	Auto	1
11/4/7	Aborted	2510	OCT 01 19:37:59 2009	F	F	Auto	1
11/4/8	Aborted	2510	OCT 01 19:38:00 2009	F	F	Auto	1
11/4/9	Aborted	2510	OCT 01 19:38:05 2009	F	F	Auto	0
11/4/10	Aborted	2510	OCT 01 19:38:10 2009	F	F	Auto	0
11/4/24	Downloaded	2510	OCT 01 23:41:57 2009	F	F	Manl	1

ONU Software Upgrades via TFTP/SNMP

Perform image downloads via TFTP rather than OMCI when CPE models and software can support it. Image upgrade through TFTP download is faster than OMCI method.

If the CPE model and its software version supports both TFTP and OMCI download method, MXK uses TFTP when it is possible.

Users can view the download method with the **onu image show** command and the **onu upgrade show** command. The following examples shows the image is download to ONU 13/2/1 with TFTP.

```
zSH> onu image download-activate-commit 13/2/1 24xx.0116
```

```
zSH> onu image show 13/2/1
```

	Partition 0	Partition 1
Version:	S3.0.116	S3.0.117
isCommitted:	True	False
isActive:	True	False
isValid:	True	True
Download status:	Complete	
Onu model id:	2426	
Upgrade start time:	OCT 17 19:19:38 2013	
Will be activated:	False	
Will be committed:	False	
Upgrade type:	Manual	
Download method:	TFTP	

```
zSH> onu upgrade show complete
```

ONU	State	Model	Start Time	Act	Cmt	Typ	Part	Method
13/2/1	Complete	2426	OCT 17 19:34:41 2013	F	F	Manl	1	TFTP
13/2/2	Complete	2426	OCT 17 20:57:03 2013	F	F	Manl	1	OMCI

Manage ONU with OMCI

This section describes how GPON ONUs are managed with OMCI (standard G988):

Topics:

- [Monitoring ONU Status and Alarms, page 910](#)
- [Rebooting, Resyncing and Reprovisioning of ONUs, page 912](#)
- [Monitoring ONU UNI ports Status and Alarms, Configuring ONU UNI port Admin Status and Port speed, page 913](#)
- [Updating the System Time on the MXK and ONUs, page 916](#)
- [Deleting ONU configuration, page 916](#)
- [Moving ONU configuration, page 919](#)
- [Cloning ONU configuration, page 920](#)

Monitoring ONU Status and Alarms

View status and alarms generated on an ONU with the `gpononu status` command.

[Table 97](#) provides the output fields description for this command.

Table 97: Gpononu status Command Output Field Explanations

Field	Description
ID	The ONU ID. In the range of 1 to 64.
Onu	The ONU interface name. By default in the format of shelf ID-Slot ID-OLT ID-ONU ID
OperStatus	The operational status of the ONU. Values: Up Down

Table 97: Gpononu status Command Output Field Explanations

Field	Description
ConfigState	The OMCI configuration states on the ONU. It is detected by the OLT side with respect to the ONU. Values: None This will probably only show during a bootup. Not yet queued for configuration. Queue Waiting to be configured. Configuring Being configured. Active configuration was successful. Inactive The ONU is down. Non-OMCI not provisioned for OMCI or SNMP. RgComError (for RG-enabled ONTs) SNMP cannot communicate with the ONT. RgServiceSetupErr (for RG-enabled ONTs) One or more SNMP commands failed. OmcErr an error occurred during the OMCI configuration. OmcErr+RgComErr both an OMCI error and SNMP communications failure. OmcErr+RgServErr both an OMCI error and failure of one or more SNMP commands.
GponOnuStatus	The standard GPON MAC alarms of the ONU detected on the OLT. Values: Active ONU is active, no alarm Inactive ONU is inactive, cannot get alarm LOS Lost of Signal LOF Lost of Frame DOW Drift of Window DG Dying Gasp SF Signal Fail SD Signal Degrade LCDG Lost of GEM Channel Delinquency RD Remote Defect RXPWRDSA Received Power of Range, and ONU is disabled TF Transmitter Failure SUF Start Up Failure LOA Lost of Ack MEM Message error PEE Physical equipment error OAML Lost of OAM

Table 97: Gpononu status Command Output Field Explanations

Field	Description
DownloadState	ONT software image download states. Values: — this ONU is not configured with OMCI None Queued NoUpgrade Downloading Complete Error Aborted
OLT Rx Power	Upstream optical power level received at the OLT.
ONT Rx Power	Downstream optical power level received at the ONU.
Distance (KM)	Calculated distance of an ONU from the OLT.

This example shows an ONU that is enabled and completes the OMCI provisioning.

```
zSH> gpononu status 4/1/1
```

ID	Onu	OperStatus	ConfigState	Download State	OLT Rx Power	ONT Rx Power	Distance (KM)	GPON OnuStatus
1	1-4-1-1	Up	Active	NoUpgrade	-19.2 dBm	-20.0 dBm	18	Active

This example shows an ONU is enabled and then goes down with a dying gasp.

```
zSH> gpononu status 4/1/1
```

ID	Onu	OperStatus	ConfigState	Download State	OLT Rx Power	ONT Rx Power	Distance (KM)	GPON OnuStatus
1	1-4-1-1	Down	Active	NoUpgrade	-19.2 dBm	-20.0 dBm	18	Inactive+LOS+LOF+DG+OAML

Rebooting, Resyncing and Reprovisioning of ONUs

This section covers the following topics:

- [Reboot an ONU, page 913](#)
- [Re-synchronize an ONU, page 913](#)
- [Re-apply an ONU, page 913](#)

The commands in the above topics are applied to both Unified Service Provisioning and Smart OMCI.

Note that the **gpononu set2default** command is related to these topics, but only applicable to the ONUs that support Residential Gateway (RG) provisioning through MXK. For the details, refer to [Set factory default for an ONU, page 856](#).

Reboot an ONU

Reboot the remote ONU from the MXK with the **gpononu reboot** command.

Reboot an ONU:

```
zSH> gpononu reboot 13/4/2
```

Re-synchronize an ONU

Synchronize an ONU with the MXK with the **gpononu resync** command. This command causes the MXK to break and re-establish linkage with ONU, and sends the previous configuration (OMCI configuration, and SNMP configuration if it applicable) to ONU. It could be used after an ME profile change in Smart OMCI configuration. It is not common to use the command in Unified Service Provisioning. It is typically only used for debug.

Re-sync an ONU:

```
zSH> gpononu resync 13/4/2
```

Re-apply an ONU

The **gpononu apply** command issues the OMCI configuration command in the ME profile. This command does not force a resync of the ONU.

If the ONU is provisioned by Smart OMCI, after users made modifications to the Specific profile or Generic profile and added new services, use the **gpononu apply** command, then these commands take effect in the ONU without affecting other existing services on the same or other ports.

If the ONU is provisioned by Unified Service Provisioning, it is not required to use the **gpononu apply** command. The new services will be updated to the ONU automatically, unless it is a post configuration in USP.

Re-apply an ONU:

```
zSH> gpononu apply 13/4/2
```

Monitoring ONU UNI ports Status and Alarms, Configuring ONU UNI port Admin Status and Port speed

This section covers the following topics:

- [Retrieve status of subscriber facing ports, page 914](#)

- [Retrieve alarm information on an ONU, page 914](#)
- [Administration of subscriber facing ports, page 914](#)
- [Configurable speed of subscriber facing ports, page 915](#)

Retrieve status of subscriber facing ports

From the MXK, the administrative state and operational state of subscriber facing ports on ONU can be retrieved by using this command:

```
gpononu status [slot[/olt[/onu]] | interfaceName] port all[portType  
[portNumber]
```

The portType is based on what are supported by the ONU model. The possible port types could be *eth* (ethernet port), *pots* (POTS port), *rf* (RF port), and *ces* (T1/E1 port).

This example shows the status of an ethernet port on ONU:

```
zSH> gpononu status 5/1/1 port eth 1
5/1/1 ONU Port Status
      Ethernet Port Status - Port 1
      Administrative State           up
      Operational State              active
```

Retrieve alarm information on an ONU

View alarms that are internal to the ONU and ONU LAN facing port with the **gpononu alarms** command.

```
zSH> gpononu alarms 13/4/2
13/4/2 ONU Active Alarms
      PptpEthUni 0x0402 LanLos
      PptpEthUni 0x0404 LanLos
```

Administration of subscriber facing ports

Enabling or disabling admin state of a subscriber facing port

From the MXK, the administrative state of an ONU subscriber facing port can be enabled or disabled by using this command:

```
gpononu port [slot[/olt[/onu]] portType interfaceID up|down
```

The portType is based on what are supported by the ONU model. The possible port types could be *eth* (ethernet port), *pots* (POTS port), *rf* (RF port), and *ces* (T1/E1 port).

Note that for Dynamic OMCI, this can also be done in CPE subscriber profiles.

- 1 Set an ONU ethernet port admin state to down:

```
zSH> gpononu port 5/1/1 eth 1 down
```

2 Verify the admin status:

```

zSH> gpononu status 5/1/1 port eth 1
5/1/1 ONU Port Status
      Ethernet Port Status - Port 1
                        Administrative State      down
                        Operational State          inactive

```

3 Set the ONU ethernet port admin state back to up:

```

zSH> gpononu port 5/1/1 eth 1 up

```

4 Verify the admin state is up:

```

zSH> gpononu status 5/1/1 port eth 1
5/1/1 ONU Port Status
      Ethernet Port Status - Port 1
                        Administrative State      up
                        Operational State          active

```

Configurable speed of subscriber facing ports

By default the GPON ONU port speed of subscriber facing ports is set to auto-detect. Users can modify this setting by using the **gpononu auto-detect** command:

```

zSh> gpononu auto-detect <slot>/<olt>/<onu> portType <interface#>
      < auto | 10F | 100F | 1000F | 10H | 100H | 1000H | 10FA | 1000A >

```

The settings for this command are:

- auto: auto-detect
- 10F: 10 Mbps, full duplex only
- 100F: 100 Mbps, full duplex only
- 1000F: 1000 Mbps, full duplex only
- 10H: 10 Mbps, half duplex only
- 100H: 100 Mbps, half duplex only
- 1000H: 1000 Mbps, half duplex only
- 10FA: 10 Mbps full duplex and auto
- 1000A: 1000 Mbps auto

Note that for Dynamic OMCI, this can also be done in CPE subscriber profiles.

The following example configures an ONU Ethernet subscriber port for 1000 full duplex mode:

```

zSH> gpononu auto-detect 10/1/1 eth 1 1000F

```

Updating the System Time on the MXK and ONUs

ONUs get time of day and time zone offset from MXK automatically with OMCI. To view the current time and date on MXK, users can use the **showdatetime** command. To view the current time zone offset, use the ntp-client-config 0 profile. The time is retrieved from one of the SNTP servers configured in the ntp-client-config 0 profile. The local-timezone field in the profile is used to set the time zone offset. The system time on the ONU maybe displayed on the CPE WEBUI, for example, zNID 24xx displays it in the System Date and Time field under the Device Info table.

To view the current time on MXK. This value will be sent to ONU by OMCI:

```
zSH> showdatetime
Current Time: TUE JAN 04 02:59:34 2013 GMT
```

To update the time settings on MXK, use the following command. The local-timezone will be sent to ONU by OMCI:

```
zSH> update ntp-client-config 0
ntp-client-config 0
Please provide the following: [q]uit.
primary-ntp-server-ip-address: ---> {0.0.0.0}: 128.2.136.71
secondary-ntp-server-ip-address: -> {0.0.0.0}: 50.31.2.213
local-timezone: -----> {gmt}: pacific CPE TimeZone Offset
daylight-savings-time: -----> {false}:
daylight-savings-time-start: -----> {}:
daylight-savings-time-end: -----> {}:
.....
Save changes? [s]ave, [c]hange or [q]uit:s
```

Deleting ONU configuration

To remove all the ONU configurations on an ONU, and set this ONU to defaults, users can use the **gpononu delete slot[/olt/onu]]command**.

The **gpononu delete** command will

1. delete all CPE subscriber profiles and CPE connections that were created on the ONU, and all CPE profiles that associated with the ONU (e.g CPE system profile, if it is configured in RG provisioning),
2. delete the ONU's OMCI Specific profile (for Smart OMCI only, if it exists),
3. delete the MXK bridges that were created on the GEM port, and GEM ports that were created on that ONU,
4. set the gpon-olt-onu-config profile of the ONU to defaults,
5. set the adminstatus, ifName, and redundancy-param1 fields in the ONU I/F translate profile to defaults.

Note that if users only want to delete the item 1 in the above list on Unified Service Provisioning ONUs, use the cpe delete slot[/olt/onu]] command. For

the details, refer to [Deleting CPE profiles and CPE connection that associated on an ONU](#) on page 790.

Deleting ONU configuration on an ONU

This example deletes the ONU configuration on ONU 1/3/1, and verifies it. This ONU is a dynamic OMCI configured zNID.

1 Remove the ONU configuration on ONU 1/3/1.

```
zSH> gpononu delete 1/3/1
```

```
Ok to delete ONU 1/3/1 and all of it's configuration? [yes] or [no]: yes
```

```
Do you want to exit from this request? [yes] or [no]: no
```

```
Are you sure? [yes] or [no]: yes
```

```
deleting ONU 1/3/1
```

```
ONU 1/3/1 has been deleted
```

2 Verify that the MXK bridges that were created on the GEM ports of ONU 1/3/1 are all gone.

```
zSH> bridge show
```

Type	Orig VLAN/SLAN	ST	Table Data	Physical	Bridge
dwn	Tagged 999			1/3/1/2/gpononu	1-3-1-257-gponport-999/
bridge	UP				
upl	ST 2/998			1/a/4/0/eth	ethernet4-2-998/bridge
UP	S SLAN 998	VLAN 2	default		
upl	ST 3/998			1/a/4/0/eth	ethernet4-3-998/bridge
UP	S SLAN 998	VLAN 3	default		
tls	Tagged 300			1/a/4/0/eth	ethernet4-300/bridge
UP					
tls	Tagged 501			1/a/4/0/eth	ethernet4-501/bridge
UP					
upl	Tagged 999			1/a/4/0/eth	ethernet4-999/bridge
UP	S VLAN 999		default		
upl	1001			1/a/8/0/eth	ethernet8/bridge
UP	S VLAN 1001		default		

7 Bridge Interfaces displayed

3 Verify that the GEM ports that were created on ONU 1/3/1 are removed.

```
zSH> gpononu gemports 1/3/1
```

Extra	Fixed UBR	Fixed CBR	Assured	Max
	traf			
Bandwidth	Bandwidth	Bandwidth	Bandwidth	Bandwidth

```

ONU          GEM Port   Admin prof    compn share Mbits/sec Mbits/sec Mbits/sec
Mbits/sec Type         allocId DBA
=====
=====
```

- 4** Verify that the CPE connections that were created on the Uni-ports of ONU 1/3/1 are removed.

```

ZSH> bridge show onu
      GEM   ONU      DSCP   ONU UNI   OLT      OLT
ONU    Port  UNI      to COS  VLAN/SLAN  VLAN/SLAN  MVR   Service OLT Bridge
      ST
-----
-----

```

- 5** Verify the CPE subscriber profiles are removed on this ONU.

```
zSH> CPE> ETH> show 1/3/1
No services found.
```

```
zSH> CPE> VOIP> show 1/3/1
No services found.
```

```
zSH> CPE> PWE> show 1/3/1
No services found.
```

- 6** To verify that the gpon-olt-onu-config profile are set to defaults, use the following command.

```
zSH> get gpon-olt-onu-config 1-1-3-1/gpononu
```

- 7 Verify that adminstatus, ifName, and redundancy-param1 fields in the if-translation profile of ONU 1/3/1 are set to default.

```
zSSH> get if-translate 1-1-3-1/gpononu
if-translate 1-1-3-1/gpononu
ifIndex: -----> {175}
shelf: -----> {1}
slot: -----> {1}
port: -----> {3}
subport: -----> {1}
type: -----> {other}
adminstatus: -----> {up}
physical-flag: -----> {false}
iftype-extension: --> {gpononu}
ifName: -----> {1-1-3-1}
redundancy-param1: -> {0}
description-index: -> {0}
```

Moving ONU configuration

The ONU move feature allows users to move the ONU configuration from one ONU to another ONU; from one OLT port to another OLT port; or from one GPON card/ slot to another GPON card/ slot. The ONU configuration here includes all the CPE subscriber profiles, CPE connections, GEM ports, bridges, and assigned serial number on the ONU. This feature could be used in many cases. For example, if the OLT SFP has some hardware failures, users can just simply unplug the fiber from this OLT port, plug-in to another OLT port, and move the ONU configuration over to the new OLT port.



Note: The destination ONU must have no ONU configuration on it.



Note: After moving ONU configuration to the destination ONU, the ONU configuration on the source ONU will be deleted.

When moving ONUs by OLT port or by GPON card, the ONU's GEM port IDs will be preserved. When moving a single ONU, the ONU may be assigned a different GEM port ID when it is moved, because another ONU may have already allocated the GEM port ID.

To move the ONU configuration from a source ONU to the destination ONU, use the **gpononu move** commands.

Command:

```
gpononu move <slot>[/<olt>[/<onu>]]to <slot>[/<olt>[/<onu>]]
```

- To move ONU configuration from one individual ONU to another:
zSH> onu move 5/1/1 to 6/2/4
- To move ONU configuration of all ONUs under one OLT port to another:
zSH> onu move 5/1 to 6/2
- To move ONU configuration of all ONUs under one GPON card to another:
zSH> onu move 5 to 6

Moving ONU configuration from one OLT to another OLT

The following example shows how to move the ONU configuration from one OLT port to another:

Move the ONU configuration from all ONUs under OLT port 5/1 to all ONUs under OLT port 5/2.

```
zSH> onu move 5/1 to 5/2
Ok to move ONUs from OLT 5/1 to OLT 5/2? [yes] or [no] :
yes
```

Do you want to exit from this request? [yes] or [no]:

no

Are you sure? [yes] or [no]: **yes**

```

querying ONU 5/2/1
querying ONU 5/2/64
querying ONU 5/2/63
querying ONU 5/2/62
querying ONU 5/2/61
querying ONU 5/2/60
:
querying ONU 5/2/4
querying ONU 5/2/3
querying ONU 5/2/2
copying ONU 5/1/1 to 5/2/1
copying ONU 5/1/64 to 5/2/64
copying ONU 5/1/63 to 5/2/63
copying ONU 5/1/62 to 5/2/62
:
copying ONU 5/1/5 to 5/2/5
copying ONU 5/1/4 to 5/2/4
copying ONU 5/1/3 to 5/2/3
copying ONU 5/1/2 to 5/2/2
deleting ONU 5/1/1
deleting ONU 5/1/64
deleting ONU 5/1/63
deleting ONU 5/1/62
:
deleting ONU 5/1/5
deleting ONU 5/1/4
deleting ONU 5/1/3
deleting ONU 5/1/2

```

ONUs from OLT 5/1 have been moved to OLT 5/2

Cloning ONU configuration

The ONU clone feature allows users to copy ONU configuration from one ONU to another one or many ONUs; from one OLT port to another one or many OLT ports; or from one GPON card/ slot to another one or many GPON cards/ slots. The ONU configuration here includes all the CPE subscriber profiles, CPE connections, GEM ports, bridges, and assigned serial number on the ONU. This feature could be used in many cases. Note that, during clone, some unique subscriber-specific fields are not copied, such as VoIP phone number, user name, password, WIFI SSID, encrypt-key, and device-pin.



Note: The destination ONU must have no ONU configuration on it. Otherwise, it will be overwritten during cloning.



Note: After cloning ONU configuration to the destination ONU, the ONU configuration on the source ONU will not be deleted.

When cloning ONUs by OLT port or by GPON card, the ONU's GEM port IDs will be preserved. When cloning a single ONU, the ONU may be assigned a different GEM port ID when it is cloned, because another ONU may have already allocated the GEM port ID.

To copy the ONU configuration from a source ONU to the destination ONU, use the **onu clone** commands.

Command:

onu clone <slot>[/<olt>[/<onu>]]to <slot>[/<olt>[/<onu>]]

<slot>, <olt>, and <onu> can be specified with ranges.

- To clone ONU configuration from one individual ONU to another ONU:

```
zSH> onu clone 5/1/1 to 6/2/4
```

- To clone ONU configuration from one individual ONU to many ONUs:

```
zSH> onu clone 5/1/1 to 6/2/[1,3-24]
```

- To clone ONU configuration of all ONUs under one OLT port to another OLT port:

```
zSH> onu clone 5/1 to 6/2
```

- To clone ONU configuration of all ONUs under one OLT port to many OLT ports:

```
zSH> onu clone 5/1 to 6/[1, 3-4]
```

- To clone ONU configuration of all ONUs under one GPON card to another GPON card:

```
zSH> onu clone 5 to 6
```

- To clone ONU configuration of all ONUs under one GPON card to many other GPON cards:

```
zSH> onu clone 5 to [6-7]
```

Cloning ONU configuration from one ONU to many ONUs

The following example shows how to clone the ONU configuration from one ONU port to many others:

Clone the ONU configuration from ONU 1/1/1 to ONUs 2/2/1, 2/2/3-2/2/24.

```
zSH> onu clone 1/1/1 to 2/2/[1,3-24]
```

```
Ok to clone ONU from 1/1/1 to 2/2/[1,3-24]? [yes] or  
[no]: yes
```

Do you want to exit from this request? [yes] or [no]:

no

Are you sure? [yes] or [no]: **yes**

deleting ONU 2/2/1

cloning ONU 1/1/1 to 2/2/1

deleting ONU 2/2/3

cloning ONU 1/1/1 to 2/2/3

deleting ONU 2/2/4

cloning ONU 1/1/1 to 2/2/4

deleting ONU 2/2/5

cloning ONU 1/1/1 to 2/2/5

deleting ONU 2/2/6

cloning ONU 1/1/1 to 2/2/6

deleting ONU 2/2/7

cloning ONU 1/1/1 to 2/2/7

deleting ONU 2/2/8

cloning ONU 1/1/1 to 2/2/8

deleting ONU 2/2/9

cloning ONU 1/1/1 to 2/2/9

deleting ONU 2/2/10

cloning ONU 1/1/1 to 2/2/10

deleting ONU 2/2/11

cloning ONU 1/1/1 to 2/2/11

deleting ONU 2/2/12

cloning ONU 1/1/1 to 2/2/12

deleting ONU 2/2/13

cloning ONU 1/1/1 to 2/2/13

deleting ONU 2/2/14

cloning ONU 1/1/1 to 2/2/14

deleting ONU 2/2/15

cloning ONU 1/1/1 to 2/2/15

deleting ONU 2/2/16

cloning ONU 1/1/1 to 2/2/16

deleting ONU 2/2/17

cloning ONU 1/1/1 to 2/2/17

deleting ONU 2/2/18

cloning ONU 1/1/1 to 2/2/18

deleting ONU 2/2/19

cloning ONU 1/1/1 to 2/2/19

deleting ONU 2/2/20

cloning ONU 1/1/1 to 2/2/20

deleting ONU 2/2/21

cloning ONU 1/1/1 to 2/2/21

deleting ONU 2/2/22

cloning ONU 1/1/1 to 2/2/22

deleting ONU 2/2/23

cloning ONU 1/1/1 to 2/2/23

deleting ONU 2/2/24

cloning ONU 1/1/1 to 2/2/24

ONU 1/1/1 has been cloned to 2/2/[1,3-24]

MXK GPON using the Reg ID for provisioning

The registration ID (Reg ID) provides an alternative, hardware-independent, method for activating an ONU. It allows service providers a flexible method for ONU replacement.

A Reg ID is a 10-digit number assigned to a customer. It is provisioned into MXK, and communicated to installation technicians to enter it into the ONU in the field.

The Reg ID process is as follows:

- When the OLT discovers a serial number it tries to match it against its provisioned serial numbers.
- If there is a serial number match, then the ONU is brought up.
- If there is no match on serial number, then:

The password (Reg ID) is retrieved and compared against the password (Reg ID) for each ONU configured with `useRegId = True`. If there is a match, the ONU is assigned the serial number and is brought up.

Configuring Reg ID

The **gpononu set** command enables users to configure the Reg ID.

```
zSH> gpononu set <slot/olt/onu> regid <xxx>
```

This command enables provisioning by Reg Id.

It sets the **password** parameter to "xxx", the **use-reg-id** parameter to **Enabled**, and the **onu-added** parameter to **True** in the **gpon-olt-onu-config** profile.

If the field already has a password (Reg ID) and it doesn't match the one in the command, the system will ask users to confirm that users want to change it.



Note: The Reg ID must be unique within the OLT.

```
zSH> gpononu clear <slot/olt/onu> regid
```

Sets **use-reg-id** to **False**, which disables auto-provisioning by Reg ID.

In addition:

- This command clears the password field to an empty string.
- If there is no serial number, then it sets the onu-added parameter to False.
- If there is a serial number, this command does not erase it or bring down the ONU.



Note:

The contents of the RegID is never displayed - it can only be viewed in the profile.

Bandwidth Allocation for Upstream Traffic from the ONU to the MXK

The bandwidth allocation for upstream traffic from the ONU to the MXK is configured in the GPON traffic profile.

This section includes the following topics:

- [Configure GPON traffic profile, page 925](#)
- [Dynamic Bandwidth Allocation \(DBA\), page 934](#)

Configure GPON traffic profile

The general function of a GPON traffic profile is to set a maximum upstream transmission rate, class of service types, and bandwidth allocation type per T-cont on the PON. When creating a GEM port with the bridge add command, users must associate a GPON traffic profile with it. To minimize the amount of configuration per subscriber, the GPON traffic profile is defined in the system, and has an index. The index of a GPON traffic profile is used in the creation of a subscriber bridge interface. Thus, if many customers want the same service, one GPON traffic profile can be created, and its index could be used for multiple GEM ports without having to define the bandwidth parameters every time a subscriber is provisioned.

For the detail DBA configuration on the GPON traffic profile, refer to section [Dynamic Bandwidth Allocation \(DBA\) on page 934](#).

The CAC validation is performed during the GPON traffic profile configuration.

Creating a GPON traffic profile

[Table 98](#) provides descriptions for GPON traffic profile parameters.

Table 98: GPON Traffic Profile parameters description

Parameter	Description
guaranteed-upstream-bw	Specifies the guaranteed non-DBA upstream bandwidth on a T-cont in Kbps. The value should be a multiple of 512 and cannot exceed 1,048,576 Kbps. Note that the guaranteed-upstream-bw is for CBR and UBR only and is not used in DBA. If DBA is disabled the guaranteed upstream bandwidth should be non-zero.
traffic-class	Specifies the upstream traffic class type of a T-cont. This is not used in DBA. Default: ubr Values: cbr Constant Bit Rate. The CBR class of traffic is used by connections that require a constant and guaranteed rate. The sampling time for CBR is constant, with no delay. ubr Unspecified Bit Rate. The UBR class of traffic does not specify traffic-related guarantees. No numerical commitments are made with respect to the packet loss or delay. With UBR service, the available bandwidth is fairly distributed to the active UBR subscribers.
compensated	CBR compensation mode. Sometimes CBR access will be skipped after OLT and ONU exchanged GPON OAM messages. If users select true in compensated mode, CBR access can be compensated immediately after exchange of the GPON OAM messages to prevent possible jitter of the CBR channel. Compensation mode is not used for DBA. Default: false Values: true false
shared	Shared feature is to let the GEM ports under the same ONU share the upstream bandwidth. Select true if the GEM port which uses this traffic descriptor shares a T-CONT (i.e. Alloc-ID) with another GEM port under the same ONU. False otherwise. Shared mode is used for both DBA and non-DBA. Default: false Values: true false

Table 98: GPON Traffic Profile parameters description

Parameter	Description
dba-enable	Enable or disable DBA. The default value is false. Default: false Values: true false
dba-fixed-us-ubr-bw	Guaranteed bandwidth in upstream direction for UBR type of traffic. It is applicable only when DBA is enabled. The minimum values of Fixed Ubr bandwidth can be 0 or 128 Kbps. The maximum value is 1,048,576 Kbps. Only multiples of 64 Kbps are allowed. Default: 0
dba-fixed-us-cbr-bw	Guaranteed bandwidth in upstream direction for CBR type of traffic. It is applicable only when DBA is enabled. The minimum value of Fixed Cbr bandwidth can be 0 or 512 Kbps. The maximum value is 454,208 Kbps. Only multiples of 64 Kbps are allowed. Default: 0
dba-assured-us-bw	DBA Assured bandwidth in upstream direction. It is applicable only when DBA is enabled. It will be allocated when traffic demand exists, it may not be given without demand. The minimum value of Assured bandwidth can be 0 or 256 Kbps. The maximum value is 1,048,576 Kbps. Only multiples of 64 Kbps are allowed. Default: 0

Table 98: GPON Traffic Profile parameters description

Parameter	Description
dba-max-us-bw	This is the maximum DBA bandwidth that can be allocated to a T-CONT. This maximum bandwidth includes guaranteed bandwidth and non-guaranteed bandwidth. Users can use this parameter to indicate the amount of non-guaranteed bandwidth configured for the traffic profile. The non-guaranteed class of service can be either nonassured or besteffort type of service, which is specified in the dba-extra-us-bw-type field. The value of the non-guaranteed bandwidth can be computed using the configured value in this parameter minus the sum of the dba-fixed-us-ubr-bw, dba-fixed-us-cbr-bw and dba-assured-us-bw. The configured value in this parameter has to be greater than or equal to sum of the dba-fixed-us-ubr-bw, dba-fixed-us-cbr-bw and dba-assured-us-bw. If the configured value is equal to the sum of the dba-fixed-us-ubr-bw, dba-fixed-us-cbr-bw and dba-assured-us-bw, then no bandwidth is assigned to non-guaranteed type of service. The maximum value is 1,048,576 Kbps. Only multiples of 64 Kbps are allowed. Default: 0
dba-extra-us-bw-type	The priority type of non-guaranteed bandwidth. Default: nonassured Values: nonassured Bandwidth only given if bandwidth is available but not guaranteed. Nonassured has higher priority for getting unused bandwidth than besteffort. besteffort Demand only met if remaining upstream bandwidth is available. Besteffort has the lowest priority.

- 1 Create a GPON traffic profile with the **new gpon-traffic-profile index** command.

The system provides the profile validation to ensure that the specified bandwidth value does not exceed the maximum value:

- If DBA is disabled, profile validation is performed to ensure that the static guaranteed-upstream-bandwidth does not exceed the maximum value.

- If DBA is enabled, profile validation is performed to ensure each DBA related bandwidth does not exceed the maximum values, and the sum of *dba-fixed-us-ubr-bw*, *dba-fixed-us-cbr-bw* and *dba-assured-us-bw* does not exceed the *dba-max-us-bw* for the GPON traffic profile.

```
mxk7-zSH> new gpon-traffic-profile 512
```

```
gpon-traffic-profile 512
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}: 2600000          in Kbps
Invalid entry: guaranteed-upstream-bw range: [0 to
1048576]
profile validation on the value range
guaranteed-upstream-bw: -> {0}: 512             in Kbps
traffic-class: -----> {ubr}: cbr
compensated: -----> {false}: true
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: -----> {0}:
dba-fixed-us-cbr-bw: -----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

- 2 View the GEM port parameter settings in a GPON traffic profile with the **get gpon-traffic-profile index** command.

```
mxk7-zSH> get gpon-traffic-profile 512
gpon-traffic-profile 512
guaranteed-upstream-bw: -> {512}
traffic-class: -----> {cbr}
compensated: -----> {true}
shared: -----> {false}
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: -----> {0}:
dba-fixed-us-cbr-bw: -----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
```

Sharing the Alloc-Id among GEM ports

The share feature in the GPON traffic profile is to let the GEM ports that under same ONU share upstream bandwidth.

The system supports up to 384 DBA Alloc-Ids per GPON physical port, and 768 Alloc-Ids per GPON physical port (including static and DBA).

Multiple GEM ports can share a single Alloc-Id if

- those GEM ports use the same traffic profile,

- the traffic profile has "shared" set to true,
- and those GEM ports are under the same ONU.

To turn on the share function, set the *shared* parameter to **true** in the GPON traffic profile.

- 1 View the Alloc-Id values assigned to the GEM ports when shared feature is disabled.

This example shows GEM ports 1-6-2-501 and 1-6-2-701 have different Alloc-Ids, 501 and 701.

```
zSH> gpononu gempports 6/2
```

Processing list of 64

This command may take several minutes to complete.

Do you want to continue? (yes or no) [no] **yes**

	traf	Fixed UBR	Fixed CBR	Assured	Max	Extra
Bandwidth	Bandwidth	Bandwidth	Bandwidth	Bandwidth	Bandwidth	Bandwidth
ONU	GEM Port	Admin	prof	compn	share	Mbits/sec
Type	allocId	DBA				
=====	=====	=====	=====	=====	=====	=====
1-6-2-1	1-6-2-501	Up	1024	False	False	1.024
501	n/a					
	1-6-2-701	Up	1024	False	False	1.024
701	n/a					
1-6-2-2	1-6-2-502	Up	1024	False	False	1.024
502	n/a					
	1-6-2-702	Up	1024	False	False	1.024
702	n/a					
Total Available BW: 1130.663(Mb), Total Available BW for Compensated CBR: 454.246 (Mb)						

- 2 Create a GPON traffic profile and enable the shared feature:

```
zSH> new gpon-traffic-profile 512
```

```
gpon-traffic-profile 512
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}: 512
traffic-class: -----> {ubr}:
compensated: -----> {false}:
shared: -----> {false}: true
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: ----> {0}:
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
```

New record saved.

- 3 Apply the GPON traffic profile to multiple GEM ports by using the **bridge add** command.
- 4 List all the ONU GEM ports use this GPON traffic profile.

```
zSH> gpononu gtp list 512
```

To Abort the operation enter Ctrl-C

GEM Ports that use Traffic Profile 512

ONU Interface	GEM Port
=====	=====
1-6-1-1	1-6-1-501
1-6-1-1	1-6-1-701
1-6-1-2	1-6-1-502
1-6-1-2	1-6-1-702

- 5 View the Alloc-Id values assigned to the GEM ports when shared feature is enabled.

This example shows GEM ports 1-6-1-501 and 1-6-1-701 have the same Alloc-Ids, 501.

```
zSH> gpononu gemports 6/1
```

Processing list of 64

This command may take several minutes to complete.

Do you want to continue? (yes or no) [no] **yes**

							traf	Fixed UBR Bandwidth	Fixed CBR Bandwidth	Assured Bandwidth	Max Bandwidth	Extra Bandwidth
ONU	GEM Port	Admin	prof	compn	share	Mbits/sec	Mbits/sec	Mbits/sec	Mbits/sec	Mbits/sec	Mbits/sec	Mbits/sec
Type	allocId	DBA										
=====	=====	=====	=====	=====	=====	=====	=====	=====	=====	=====	=====	=====
1-6-1-1	1-6-1-501	Up	512	False	True	0.5	0	n/a	n/a	n/a	n/a	n/a
501	n/a											
	1-6-1-701	Up	512	False	True	0.5	0	n/a	n/a	n/a	n/a	n/a
501	n/a											
1-6-1-2	1-6-1-502	Up	512	False	True	0.5	0	n/a	n/a	n/a	n/a	n/a
502	n/a											
	1-6-1-702	Up	512	False	True	0.5	0	n/a	n/a	n/a	n/a	n/a
502	n/a											

Modifying a GPON traffic profile

Modify a GPON traffic profile with the **update gpon-traffic-profile index** command. Users can only modify a GPON traffic profile that is not being used by a GEM port.

Modify a GPON traffic profile:

```
mxk7-zSH> update gpon-traffic-profile 512

gpon-traffic-profile 512
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {512}:
traffic-class: -----> {cbr}: ubr
compensated: -----> {true}: false
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: ----> {0}:
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

The profile validation checks to see if the profile is being used by an ONU GEM port. A GPON traffic profile is considered as “in-use” if it is already assigned to a GEM port. If a GPON traffic profile is in-use, the GPON traffic profile modification is rejected, and an error message appears.

```
mxk7-zSH> update gpon-traffic-profile 513
gpon-traffic-profile 513
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {512}:
traffic-class: -----> {cbr}: ubr
compensated: -----> {true}: false
shared: -----> {false}:
dba-enabled: -----> {false}:
dba-fixed-us-ubr-bw: ----> {0}:
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: -----> {0}:
dba-max-us-bw: -----> {0}:
dba-extra-us-bw-type: ---> {nonassured}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
Profile Validation Error. The GTP profile is in use
and cannot be modified.
```

Starting over....

Deleting a GPON traffic profile

Delete a GPON traffic profile with the **delete gpon-traffic-profile** *index* command. Users can only delete a GPON traffic profile that is not being used by a GEM port.

Delete a GPON traffic profile:

```
mxk7-zSH> delete gpon-traffic-profile 512
```



```
gpon-traffic-profile 512
1 entry found.
```

```
Delete gpon-traffic-profile 512? [y]es, [n]o, [q]uit: y
gpon-traffic-profile 512 deleted.
```

If a GPON traffic profile is in use, the deletion will be rejected by the profile validation, and an error message will appear.

```
zSH> delete gpon-traffic-profile 514
gpon-traffic-profile 514
1 entry found.
Delete gpon-traffic-profile 514? [y]es, [n]o, [q]uit :
yes
Profile Validation Error. The GTP profile is in use
and cannot be deleted.
gpon-traffic-profile 514 not deleted.
```

Viewing the existing GPON traffic profiles

View the existing GPON traffic profiles in the system with the **list gpon-traffic-profile** command.

```
mxk7-zSH> list gpon-traffic-profile
gpon-traffic-profile 1
gpon-traffic-profile 1024
gpon-traffic-profile 2
```

Viewing the GEM ports that use the same GPON traffic profile

View the GEM ports that use the same GPON traffic profile with the **gpononu gtp list GTPId** command.

```
zSH> gpononu gtp list 512
To Abort the operation enter Ctrl-C
GEM Ports that use Traffic Profile 512
ONU Interface      GEM Port
=====
1-13-1-1          1-13-1-501
1-7-7-3           1-7-7-503
1-7-7-4           1-7-7-504
1-7-7-5           1-7-7-505
```

Modifying the GPON traffic profile index of a GEM port

The GTP index assigned to a GEM port can be modified by the **update gpon-port-config** command. The new GTP index is specified in the **traffic-profile** parameter.

When modifying a GTP index of a GEM port, the profile validation compares the new GTP with the GTP assigned on the GEM port. The GTP index modification is rejected when any of the following conditions are met:

- If the DBA is disabled on both GTP profiles, and Traffic Class is changed.

The error message “Profile Validation Error: Cannot change the GTP index as this causes a change of Class of Service” appears.

- If the status of DBA is changed.

The error message "Profile Validation Error: Cannot change the GTP index as this causes change of DBA status." appears.

If the profile validation is successful, then the CAC validation is performed. The total available bandwidth is recalculated on the GPON physical port using the newly assigned GTP index. If the CAC validation fails, the error message "CAC Validation Error: The total available bandwidth exceeded" appears.

```
zSH> update gpon-port-config 1-13-1-501/gponport
Please provide the following: [q]uit.
multicast: -> {false}:
encrypted: -> {true}:
direction: -> {bidirectional}:
traffic-profile: -> {1}:512
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Dynamic Bandwidth Allocation (DBA)

Dynamic Bandwidth Allocation (DBA) is specified in the ITU standard 984.3. This feature is used to grant upstream bandwidth to ONUs based on their demand and service-level agreement. The OLT will grant ONUs an increase in their slot time for more bandwidth while granting a reduced slot time to others. Through DBA, a GPON link can be oversubscribed for upstream traffic, and improve bandwidth usage efficiency.

Enabling and Configuring DBA bandwidth on T-ports

User can enable or disable DBA and configure the DBA bandwidth on each T-port with the GPON traffic profile.

User can configure both guaranteed (i.e. dba-fixed-us-ubr-bw, dba-fixed-us-cbr-bw, dba-assured-us-bw) and non-guaranteed DBA bandwidth (i.e. dba-max-us-bw). The non-guaranteed DBA bandwidth is always the UBR type of traffic and is non-compensated.

For the description of the DBA related GPON GEM ports uplink parameters, refer to [Table 98 on page 926](#).

- 1 Create a GPON-traffic-profile with DBA-related parameters. DBA is enabled in this GTP:

```
zSH> new gpon-traffic-profile 430080
gpon-traffic-profile 430080
```

```
Please provide the following: [q]uit.
guaranteed-upstream-bw: -> {0}: For CBR and UBR only, not
used in DBA.
traffic-class: -----> {ubr}:For CBR and UBR
only, not used in DBA
compensated: -----> {false}:Not used in DBA
shared: -----> {false}:For DBA and non-DBA
dba-enabled: -----> {false}:true DBA only
dba-fixed-us-ubr-bw: ----> {0}:512 DBA only
dba-fixed-us-cbr-bw: ----> {0}: DBA only
dba-assured-us-bw: ----> {0}:1024 DBA only
dba-max-us-bw: -----> {0}:1024000 DBA only
dba-extra-us-bw-type: ---> {nonassured}: DBA only
.....
Save changes? [s]ave, [c]hange or [q]uit:s
New record saved.
```

2 Verify the settings:

```
zSH> get gpon-traffic-profile 430080
gpon-traffic-profile 430080
guaranteed-upstream-bw: -> {0}:
traffic-class: -----> {ubr}:
compensated: -----> {false}:
shared: -----> {false}:
dba-enabled: -----> {true}:
dba-fixed-us-ubr-bw: ----> {512}:
dba-fixed-us-cbr-bw: ----> {0}:
dba-assured-us-bw: ----> {1024}:
dba-max-us-bw: -----> {1024000}:
dba-extra-us-bw-type: ---> {nonassured}:
```

3 Apply the DBA enabled GTP to an GEM port.

```
zSH> bridge add 1-3-1-5/gpononu gem 501 gtp 430080 downlink vlan 100 tagged
Adding bridge on 1-3-1-5/gpononu
Created bridge-interface-record 1-3-1-501-gponport-100/bridge
```

4 View the DBA bandwidth settings on GEMs on an ONU.

The allocID and DBA type will not be displayed until the ONU is activated.

```
zSH> gpononu gemports 3/1/5
```

				Fixed UBR	Fixed CBR	Assured	Max
Extra		traf		Bandwidth	Bandwidth	Bandwidth	
ONU	GEM Port	Admin prof	compn share	Mbits/sec	Mbits/sec	Mbits/sec	
Mbits/sec	Type	allocId	DBA				
1024	Nonassured	56	SR	0.512	0	1.024	

Changing the DBA type per ONU

There are two types of DBA, Status Reporting (SR) and Non-Status Reporting (NSR):

- SR: The ONU provides the bandwidth status information as part of the upstream traffic message. SR is specified in the Dynamic Bandwidth Report upstream (DBRu).
- NSR: NSR is the non-status reporting option where the OLT calculates the available bandwidth. The allocation is based on monitoring ONU's bandwidth usage compared to the allocated bandwidth.

By default, DBA type on each ONU is NSR. Users can change the DBA type to SR only if the ONU is inactive.



Note: Before changing the DBA type of an ONU from the default type NSR to SR, make sure the ONU supports SR.



Note: The only way to change the DBA type on an activated ONU is to clear and re-activate the ONU for the change to take effect.

This example changes DBA type of an activated ONU from NSR to SR.

1 View the DBA type on GEMs on an activated ONU.

```
zSH> gpononu gemports 3/1/5
```

				Fixed UBR	Fixed CBR	Assured	Max
Extra		traf		Bandwidth	Bandwidth	Bandwidth	
ONU	GEM Port	Admin prof	compn share	Mbits/sec	Mbits/sec	Mbits/sec	
Mbits/sec	Type	allocId	DBA				
1024	Nonassured	1-3-1-501	Up 430080	False	False	0.512	0 1.024
		56	SR				

2 Enable the DBA type to SR on the ONU.

```
zSH> update gpon-olt-onu-config 1-3-1-5/gpononu
gpon-olt-onu-config 1-3-1-5/gpononu
Please provide the following: [q]uit.
serial-no-vendor-id: -----> {ZNTS}: **
read-only **
serial-no-vendor-specific: -----> {0}: **
read-only **
password: -----> {}:
auto-learn: -----> {enabled}:
power-level: -----> {0}:
us-ber-interval: -----> {5000}:
ds-ber-interval: -----> {5000}:
onu-added: -----> {false}:
omci-file-name: -----> {}:
ONU-Managed-Entity-Profile-name: -----> {}:
```

```

ONU-Generic-Assignments-Profile-name: -> {}:
physical-traps: -----> {disabled}:
ont-traps: -----> {disabled}:
line-status-traps: -----> {disabled}:
auto-upgrade: -----> {enabled}:
serial-no-vendor-specific-fsan: -----> {0}: **
read-only **
use-reg-id: -----> {disabled}:
us-rx-power-monitoring-mode: ----->
{monitoronly}:
us-rx-power-high-threshold: -----> {-10}:
us-rx-power-low-threshold: -----> {-30}:
dba-status-reporting: ----->
{disabled}:enabled
.....
Save changes? [s]ave, [c]hange or [q]uit: s
New record saved.

```

3 De-register this ONU.

```
zSH> gpononu clear 3/1/5
```

4 Display the ONUs currently on the OLT, and discover the available serial numbers.

```
zSH> gpononu show 3/1
```

```
Free ONUs for slot 3 olt 1:
```

1	2	3	4	5	6	7	8	9	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64								

```
Discovered serial numbers for slot 3 olt 1:
```

sernoID	Vendor	Serial Number	Time Discovered
1	ZNTS	138543368	JUL 10 01:11:18 2014

5 Activate this ONU.

```
zSH> gpononu set 3/1/5 1
```

```
Onu 5 successfully enabled with serial number ZNTS
138543368
```

GEM port creation

This section includes the following topics:

- [Create a GEM port, page 938](#)
- [View the GEM port-related information, page 941](#)
- [Locate the ONU with its GEM port, page 942](#)
- [GEM port level encryption, page 943](#)

Create a GEM port

GEM ports could be created by “bridge add” commands. This section describes how to create GEM port on the bridge for data, voice, and video services.

Creating bridges on GEM ports for data, voice, and video services

This procedure describes how to use the **bridge add** command to create bridges on GEM ports to pass traffic between the MXK and the downlink ONUs for triple-play service. For different services, users can associate different GPON traffic profile with the GEM port.



Note: When creating multiple VLANs on same GEM port, the GTP must be the same. Otherwise the command will be rejected.

This section also describes how to configure an uplink bridge to pass traffic between the MXK and the upstream data/voice/video source.

Before creating a GEM port, users must create a GPON Traffic Profile. The GTP provides the rate limiting on the T-cont where the GEM port is connected. For details on creating a GTP, refer to [Configure GPON traffic profile on page 925](#). The following examples show that GEM port 501 is configured for data service, and associated with GPON traffic profile 1; GEM port 701 is configured for voice service, and associated with GPON traffic profile 2; GEM port 901 is configured for video service, and associated with GPON traffic profile 3.

The ONU in this example is managed with Smart OMCI, so the GEM index 5xx, 7xx, and 9xx match the GEM index that is selected in the Smart OMCI web-interface.

For more information on how to configure video bridging, see [Video Configuration on page 427](#).

1 Create a bridging configuration for data services:

```
zSH> bridge add 1-a-2-0/eth uplink vlan 100                                uplink bridge

zSH> bridge add 1-13-1-501/gponport gtp 1 downlink vlan 100 tagged downlink
GEM port
```

2 Create a bridging configuration for voice services:

```
zSH> bridge add 1-a-2-0/eth uplink vlan 200 uplink bridge
```

```
zSH> bridge add 1-13-1-701/gponport gtp 2 downlink vlan 200 tagged downlink
GEM port
```

3 Create a bridging configuration for video services:

```
zSH> bridge add 1-a-2-0/eth uplink vlan 300 tagged igmpproxy uplink bridge
```

```
zSH> bridge-path modify 1-a-2-0-eth-300/bridge vlan 300 default igmptimer
30 igmpsnooping enable uplink bridge path
```

```
zSH> bridge add 1-13-1-901/gponport gtp 3 downlink vlan 300 tagged video 1/
6 downlink GEM port,
the video keyword and multicast-control-list/multicast-control-entries has to be specified. If multicast-control-list
1 has no entries, this bridge will fail to pass any video traffic. If specifying 0/6 instead of 1/6, the bridge will pass
all IP multicast.
```

4 View the newly created GEM ports and associated traffic profiles for selected ONU with the **gpononu gempports** command.

```
zSH> gpononu gempports 13/1/1
```

		traf		Fixed UBR	Fixed CBR	Assured	Max	Extra
				Bandwidth	Bandwidth	Bandwidth	Bandwidth	
Bandwidth								
ONU	GEM Port	Admin	prof	compn	share	Mbits/sec	Mbits/sec	Mbits/sec
Type	allocId	DBA						
=====	=====	=====	=====	=====	=====	=====	=====	=====
1-13-1-1	1-13-1-501	Up	1	False	False	0.512	0	n/a
501	n/a							
	1-13-1-701	Up	2	True	False	0	0.512	n/a
701	n/a							
	1-13-1-901	Up	3	True	False	0	0.512	n/a
901	n/a							

Connection Admission Control (CAC) validation during the bridge/interface creation on a GEM port

When using the **bridge add** command to add a GEM port, or add a bridge to an existing GEM port with different VLAN, the following Connection Admission Control (CAC) validations are performed:

- Total available GEM ports should not exceed the maximum GEM ports.
Each ONU can support maximum of 16 GEM ports. If the value is greater then the error message "CAC Validation Error: The maximum allowed GEM ports of <value> exceeded" appears.

- Total available Alloc-Ids should not exceed the allowed values. The maximum # of Alloc-ids for Dynamic Bandwidth Allocation (DBA) is 384. The total # of Alloc-Id allowed is 768 (includes non-DBA and DBA).

If the validation fails, the error message "CAC Validation Error: The maximum allowed Alloc-ids <value> exceeded" appears. If the user tried to configure DBA, then 384 is displayed as the <value>. If the user tried to configure non-DBA, then 768 is displayed as the <value>.

- The guaranteed, assured, and fixed class of services on the GPON physical port (i.e. OLT interface) should not exceed the allowable bandwidth. The allowable bandwidth is the PON overhead subtracted from 1.248Gig per GPON physical port for upstream bandwidth. The average PON overhead is 110M.
 - The total bandwidth on all the GEM ports on the GPON physical port should not exceed the total available bandwidth.
- If the validation fails, the error message "CAC Validation Error: The total available bandwidth was exceeded" appears.
- Each GPON physical port can support 454246 Kbps available bandwidth for CBR. This bandwidth will be reduced by the UBR allocations that exceed 681,370 kbps.
 - There is a 5% overhead for all DBA bandwidth allocations.
 - Enabling OLT US FEC Parity will decrease available bandwidth by 145Mb/sec.

Before using the **bridge add** command to create a GEM port, users can use the following two commands to check the available Alloc-Ids and available bandwidth on the GPON physical port.

- 1 Check the available Alloc-Ids on a GPON physical port with the **gponolt status gtp** command:

```
zSH> gponolt status gtp
```

OLT Interface	OLT State	# GEM Ports	DBA		Total		# ONUs configured	# ONUs active
			used	avail	used	avail		
6/1	Active	0	0	384	0	768	0	0
6/2	Inactive	0	0	384	0	768	0	0
6/3	Inactive	0	0	384	0	768	0	0
6/4	Inactive	0	0	384	0	768	0	0
6/5	Inactive	0	0	384	0	768	0	0
6/6	Inactive	0	0	384	0	768	0	0
6/7	Inactive	0	0	384	0	768	0	0
6/8	Inactive	0	0	384	0	768	0	0

It also shows the OLT state. The possible values of the OLT state are:

- Active: SFP is connected, fiber is connected, and active ONU is connected.
- Ready: SFP is connected but no light seen on fiber.
- Inactive: No SFP connected.

2 Check the available bandwidth on a GPON physical port with the **gponolt show bw** command:

```
zSH> gponolt show bw 7/7
SLOT 7/OLT 7:
Total Available BW..... 1090560 Kbps
Total Available BW for Compensated CBR... 454246 Kbps
Allocated UBR BW..... 32768 Kbps
Allocated CBR BW..... 0 Kbps
Allocated Compensated CBR BW..... 0 Kbps
Allocated Assured BW..... 0 Kbps
Allocated Non-Assured BW..... 0 Kbps
Allocated Best-Effort BW..... 0 Kbps
```

View the GEM port-related information

View the GEM port related information with the **gpononu gemports** command.

```
zSH> gpononu gemports 7/3/1
```

		traf	Fixed UBR Bandwidth	Fixed CBR Bandwidth	Assured Bandwidth	Max Bandwidth	Extra Bandwidth
ONU	GEM Port	Admin	prof	compn	share	Mbits/sec	Mbits/sec
allocId	DBA						Type
=====	=====	=====	=====	=====	=====	=====	=====
=====	=====	=====					
1-7-3-1	1-7-3-501	Up	1024	False	False	1.024	0
n/a	501	n/a					n/a
	1-7-3-901	Up	512	False	False	0.512	0
n/a	640	n/a					n/a

Table 99: Gpononu gemports Command Output Field Explanations

Field	Description
Onu	The ONU interface name in the format of shelf ID-Slot ID-OLT ID-ONU ID.
GEM Port	The ONU GEM port name in the format of shelfID-SlotID-OLT ID-ONU GEM Port ID.
Admin	The administrative status of the ONU
traf prof	The traffic profile index applied to the GEM port.

Table 99: Gpononu gemparts Command Output Field Explanations

Field	Description
Compn	The compensation mode specified in the GPON traffic profile of the GEM port. Values: True False
Share	The shared mode specified in the GPON traffic profile of the GEM port. Values: True False
Fixed UBR Bandwidth Mb/s/sec	Fixed UBR bandwidth used when DBA is enabled.
Fixed CBR Bandwidth Mb/s/sec	Fixed CBR bandwidth used when DBA is enabled.
Assured Bandwidth Mb/s/sec	DBA Assured bandwidth will be allocated when traffic demand exists.
Max Bandwidth Mbit/sec	Use this parameter to indicate the amount of non-guaranteed bandwidth configured for the traffic profile. Only available when DBA is enabled.
Extra Bandwidth Type	The priority type of non-guaranteed bandwidth. Only available when DBA is enabled.
allocID	The Alloc-Id assigned on this GEM port. If DBA is enabled, then this Alloc-Id is DBA enabled Alloc-Id, otherwise it is non-DBA Alloc-Id.
DBA	The DBA type. Values: SR indicates the DBA type is Status Reporting and there is no error NSR indicates the DBA type is Non- Status Reporting and there is no error. NSR is the default value. NSR-Error indicates the DBA type NSR and there is an error in either getting the report from the ONU or the ONU does not support NSR. n/a indicates DBA has been disabled on the GEM port, or cannot communicate with ONU, or ONU has not been added. Error Indicates there are some errors

Locate the ONU with its GEM port

GEM ports can be located by using the **onu find gem** command. the **olt** field must be specified.

```
zSH> onu find gem 501 olt 4/1
GEM Port ID 501 on OLT 4/1 is allocated to ONU 4/1/1
```

GEM port level encryption

GPON OLT supports Advanced Encryption Standard (AES) 128 bits encryption transmission in downstream direction. The AES key exchange and switch-over is initiated from GPON system periodically.

By default, the encryption on GEM port level is disabled. To enable the encryption on the GEM port, users can specify **encrypted** keyword in the **bridge add** command when creating this GEM port. Note that the upstream data is not encrypted. To be able to use encryption on the downstream data, users must enable the key-exchange in the gpon-olt-config profile as well.

Enabling the encryption on the GEM ports

- 1 Create a GEM port and enable the encryption on the GEM port.

```
zSH> bridge add 1-1-1-3/gpononu gem 303 encrypted gtp 1 downlink vlan 203 tagged
Adding bridge on 1-1-1-3/gpononu
Created bridge-interface-record 1-1-1-303-gponport-203/bridge
```

- 2 Show the GEM port is set to encrypted now:

```
zSH> get gpon-port-config 1-1-1-303/gponport
gpon-port-config 1-1-1-303/gponport
Please provide the following: [q]uit.
multicast: -----> {false}:
encrypted: -----> {true}:
direction: -----> {bidirectional}:
traffic-profile: -----> {0}:
traffic-mngt-profile: -> {0}:
```

- 3 Enable the key exchange on the OLT port:

```
zSH> update gpon-olt-config 1-1-1-0/gponolt
gpon-olt-config 1-1-1-0/gponolt
Please provide the following: [q]uit.
max-rt-propagation-delay: -----> {200}:
max-onu-response-time: -----> {50}:
preassigned-eqd: -----> {0}:
los-alpha: -----> {4}:
lof-alpha: -----> {4}:
loam-alpha: -----> {3}:
scrambler: -----> {enabled}:
fec-mode: -----> {disabled}:
auto-learn: -----> {enabled}:
power-level: -----> {0}:
guard-bit-count: -----> {32}:
dba-mode: -----> {predictive}:
gem-block-size: -----> {16}:
us-ber-interval: -----> {5000}:
ds-ber-interval: -----> {5000}:
ber-sf-threshold: -----> {3}:
ber-sd-threshold: -----> {5}:
fec-request: -----> {disabled}:
```

```

key-exchange: -----> {disabled}:enabled
min-rt-propagation-delay: -----> {0}:
min-onu-response-time: -----> {10}:
eqd-measure-cycles: -----> {5}:
drift-ctrl-interval: -----> {1000}:
drift-ctrl-limit: -----> {3}:
alloc-cycle-length: -----> {2}:
min-us-alloc: -----> {16}:
ack-timeout: -----> {2000}:
pls-max-alloc-size: -----> {120}:
dba-cycle: -----> {2}:
sr-dba-reporting-block-size: --> {48}:
protection-switchover-timer: --> {500}:
preamble-override: -----> {disabled}:
preamble-type-0: -----> {0x00}:
preamble-type-1: -----> {0x00}:
preamble-type-3-pre-range: ----> {0x0b}:
preamble-type-3-post-range: ---> {0x08}:
preamble-type-3-pattern: -----> {0xaa}:
bip-error-monitoring-mode: ----> {monitoronly}:
errors-per-sample-threshold: --> {100}:
errored-samples-threshold: ----> {10}:
bip-max-sample-gap: -----> {10}:
rogue-onu-detection: -----> {disabled}:
rogue-onu-detect-frequency: ---> {10}:
rogue-onu-rx-power-threshold: -> {-30}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s

```

GPON ONU serial number format (Hexadecimal or Decimal)

By default, the GPON ONU serial number is displayed in hexadecimal. It could also be displayed in decimal.

Display the available ONUs and the discovered serial numbers in hex format with the **gpononu show slot/olt** command:

```
zSH> gpononu show 5/1
Processing list of 64
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
Free ONUs for slot 5 olt 1:
   2    3    4    5    6    7    8    9   10   11   12   13
  14   15   16   17   18   19   20   21   22   23   24   25
  26   27   28   29   30   31   32   33   34   35   36   37
  38   39   40   41   42   43   44   45   46   47   48   49
  50   51   52   53   54   55   56   57   58   59   60   61
  62   63   64
Discovered serial numbers for slot 5 olt 1:
sernoID  Vendor  Serial Number      Time Discovered
   2      ZNTS    00F1B37F          JUL 10 01:11:18 2014
   3      ZNTS    84200459          JUL 10 01:11:18 2014
```

Display the serial number in decimal format with the **gpononu show slot/olt -d** command:

```
zSH> gpononu show 5/1 -d
Processing list of 64
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
Free ONUs for slot 5 olt 1:
   2    3    4    5    6    7    8    9   10   11   12   13
  14   15   16   17   18   19   20   21   22   23   24   25
  26   27   28   29   30   31   32   33   34   35   36   37
  38   39   40   41   42   43   44   45   46   47   48   49
  50   51   52   53   54   55   56   57   58   59   60   61
  62   63   64
Discovered serial numbers for slot 5 olt 1:
sernoID  Vendor  Serial Number      Time Discovered
   2      ZNTS    15840127          JUL 10 01:11:18 2014
   3      ZNTS    2216690777        JUL 10 01:11:18 2014
```

Display the serial number in decimal format with the **gpononu showall slot/olt -d** command:

```
zSH> gpononu showall 7/7 -d
Processing list of 64
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
```

ONU	Name	Enabled	Model #	Serial Number	OMCI files and profiles
===	=====	=====	=====	=====	=====

1	1-7-7-1	Yes	2510	ZNTS 1341	ME 2210-me GEN 2210-gen
2	1-7-7-2	Yes		ZNTS 1405	ME 2210-me GEN 2210-gen
3	1-7-7-3	Yes		ZNTS 1263	ME 2210-me GEN 2210-gen
4	1-7-7-4	Yes		ZNTS 1359	ME 2210-me GEN 2210-gen
5	1-7-7-5	Yes		ZNTS 1285	ME 2210-me GEN 2210-gen
6	1-7-7-6	Yes		ZNTS 1387	ME 2210-me GEN 2210-gen
7	1-7-7-7	Yes		ZNTS 1335	ME 2210-me GEN 2210-gen
8	1-7-7-8	Yes		ZNTS 1371	ME 2210-me GEN 2210-gen

<SPACE> for next page, <CR> for next line, A for all, Q to quit

Associate a vendor ID and a serial number with an ONU when activating the ONU

Enable an ONU with the vendor ID and serial number by using the **gpononu set slot/olt/onu vendorid vendorId serno [fsan a hex number] | [a decimal number]** command. Users can specify serial number in hex or decimal format. **fsan** indicates the serial number is in hex format.

Usually the vendor ID and serial number can be found in a sticker on the ONU. For example, a small sticker on an ONU 2510 shows the FSAN serial number, e.g. FSAN-**ZNTS00F1B37F**. The first four characters, ZNTS, are vendor specific ID, and the following characters, 00F1B37F, are serial number in hex format.



Note: Attempting to provision an ONU that has a vendor ID other than ZNTS will not complete successfully and the system will return an error message, "Alert!! Foreign ONT detected - cannot activate.". Zhone requires official ONT interop for third party ONTs.

Associate a vendor ID and a hex serial number with an ONU and enable this ONU:

```
zSH> gpononu set 5/1/2 vendorid ZNTS serno fsan 00F1B37F
Onu 2 successfully enabled with serial number ZNTS 00F1B37F
```

Associate a vendor ID and a decimal serial number with an ONU and enable this ONU:

```
zSH> gpononu show 5/1 -d
Processing list of 64
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
Free ONUs for slot 5 olt 1:
    3    4    5    6    7    8    9   10   11   12   13   14
```

15	16	17	18	19	20	21	22	23	24	25	26
27	28	29	30	31	32	33	34	35	36	37	38
39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62
63	64										

Discovered serial numbers for slot 5 olt 1:

sernoID	Vendor	Serial Number	Time Discovered
3	ZNTS	2216690777	JUL 10 01:11:18 2014

zSH> **gpononu set 5/1/3 vendorid ZNTS serno 2216690777**

Onu 3 successfully enabled with serial number ZNTS 2216690777

Received Signal Strength Indication (RSSI) and Digital Diagnostic Monitoring (DDM)

Viewing the OLT and ONU optical power

Received Signal Strength Indication (RSSI) is the capability of a SFP by which the SFP reads the strength of the signal received on the both OLT and ONU side.

The user can view the upstream optical power level received at the OLT, and the downstream optical power level received at the ONU.

Note that the downstream optical power level received at the ONU (i.e. ONT Rx Power or ONT Receive Power) should be -28 or above for SFP-B+.

By default, if the measured upstream optical power of the ONU received at the OLT (i.e. OLT Rx Power or OLT Receive Power) is beyond the value range of -10 dBm to -30 dBm, the MXK will trigger a local alarm, and send a trap to ZMS. For the detail, refer to [GPON High and Low Receive Power Threshold Alarms on page 992](#).

- 1 The following example shows the ONT receive power on 1/7/3/1 is -23.0 dBm, in the normal range.

```
zSH> gpononu power show 7/3
Processing list of 64
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
```

	OLT	ONT
Interface	Receive Power	Receive Power
=====	=====	=====
1-7-3-1	-17.4 dBm	-23.0 dBm
1-7-3-2	-17.2 dBm	-23.0 dBm
1-7-3-3	-18.2 dBm	-23.0 dBm
1-7-3-4	-18.0 dBm	-23.0 dBm
1-7-3-5	-17.4 dBm	-23.0 dBm
1-7-3-6	-17.8 dBm	-24.0 dBm
1-7-3-7	-16.9 dBm	-23.0 dBm
1-7-3-8	-17.4 dBm	-23.0 dBm
1-7-3-9	-17.3 dBm	-23.0 dBm
1-7-3-10	-17.2 dBm	-22.0 dBm
1-7-3-11	error	-25.0 dBm
1-7-3-12	NA	NA

```
<SPACE> for next page, <CR> for next line, A for all, Q
to quit q
```

If there is no SFP inserted in the OLT, or the OLT/ ONU admin status is set to down, then its Receive Power field displays the value “NA”.

If the Receive Power field displays the value “error”, it means the measurement failed. Users can run the **gpononu power show** command again.

- 2 The **gpononu status** command can display the same information. It displays upstream optical power level received at the OLT in the *OLT Rx Power* column and downstream optical power level received at the ONU in the *ONT Rx Power* column.

```
zSH> gpononu status 7/3/1
```

```

                                Download   OLT       ONT       Distance  GPON
ID    Onu   OperStatus ConfigState State     Rx Power  Rx Power  (KM)   OnuStatus
==  =====
1    1-7-3-1   Up       Active    NoUpgrade -17.4 dBm -23.0 dBm  0.0    Active

```

Viewing the transmit parameters on OLT

Digital Diagnostic Monitoring (DDM) provides diagnostic information about the SFP. With DDM function, the SFP optical transceiver measures the transceiver temperature, transceiver supply voltage, Tx Bias current, and Tx output power parameters on an OLT, and also reports an End of Life status. An alarm is raised when an End of Life condition is reached.

Perform DDM on the GPON OLT card with the **gponolt show port** [slot [/olt]] command.

Table 100 provides the output fields description for this command.

Table 100: Gponolt port show Command Output Field Explanations

Field	Description
<i>SLOT/OLT</i>	The GPON ID.
Temperature	Internally measured Transceiver Temperature of the OLT in Celsius.
Voltage	Internally measured Transceiver Supply Voltage of the OLT in Volts.
Tx Bias Current	Measured Tx Bias current per OLT in Milli Amperes.
Tx Power	Measured Tx Output Power of the OLT in dBm.
End of Life Status	When the End Of Life (EOL) Alarm bit is set an alarm will be raised. SFP automatically maintains a laser output optical power by adjusting the laser current. Alarm is raised when the SFP reaches the end of life which is about 150% of original current. Alarm will be cleared when the SFP is connected. The alarm severity level is Major. Values: ok No alarm conditions are raised warning Warning is set when EOL is at about 130% original current. alarm Alarm conditions are raised SFP not present SFP is not detected

```
zSH> gponolt show port
```

```

SLOT/OLT Temperature Voltage Tx Bias Current Tx Power End Of Life Status
=====
7/1          NA        NA          NA          NA        SFP not present

```

MXK GPON Cards

7/2	NA	NA	NA	NA	SFP not present
7/3	43c	3.3v	10mA	3.0dBm	Ok
7/4	NA	NA	NA	NA	SFP not present
7/5	NA	NA	NA	NA	SFP not present
7/6	NA	NA	NA	NA	SFP not present
7/7	NA	NA	NA	NA	SFP not present
7/8	NA	NA	NA	NA	SFP not present
SLOT/OLT	Temperature	Voltage	Tx Bias Current	Tx Power	End Of Life Status
=====	=====	=====	=====	=====	=====
11/1	46c	2.9v	22mA	2.1dBm	Ok
11/2	NA	NA	NA	NA	SFP not present
11/3	NA	NA	NA	NA	SFP not present
11/4	35c	3.3v	25mA	4.1dBm	Ok

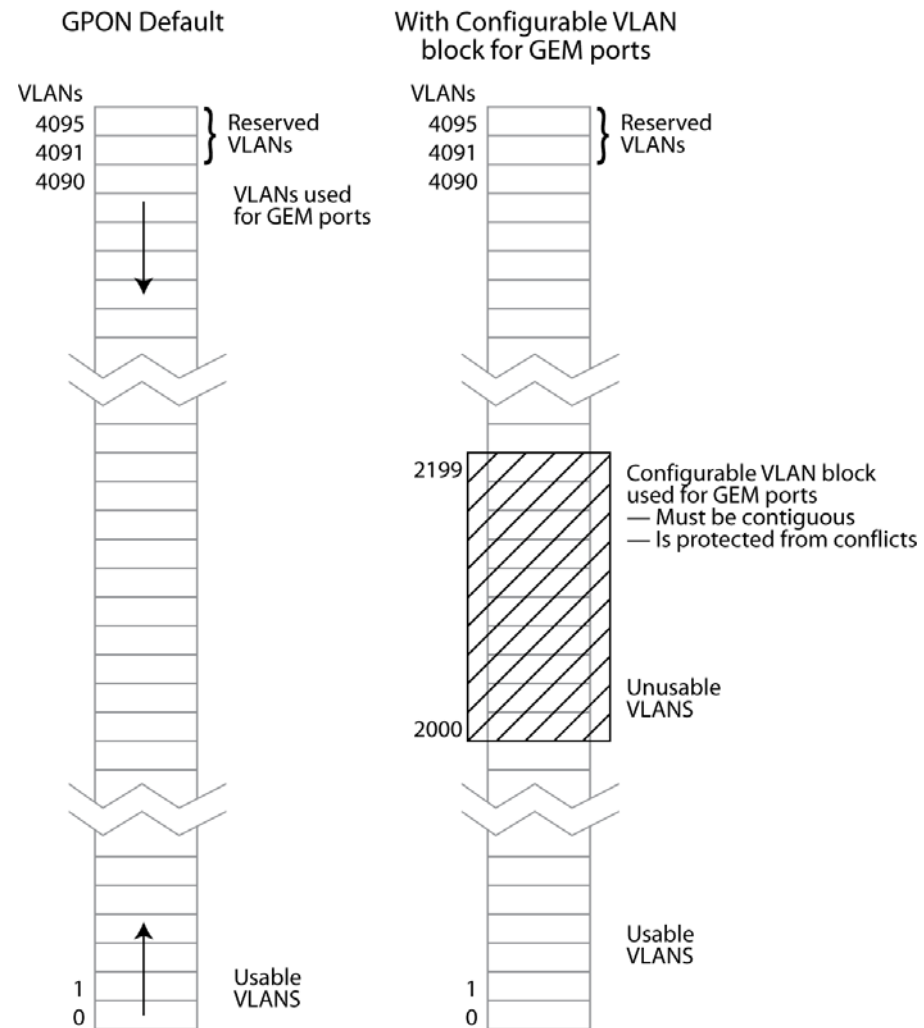
Configurable range for Reserved VLAN per GEM port

GPON line cards in an MXK chassis must reserve and assign one VLAN to each GEM port. This reserved VLAN guides traffic from the local host to its associated GEM port and is different than the VLAN you would assign in a bridge interface. The reserved VLAN per GEM port is assigned by the system when the first bridge interface is created on a GEM port and the binding persists until the last interface is removed from the GEM port.

The VLANs used for guiding GEM ports come from the same pool of VLANs which are used for other traffic, data, video or voice. The reserved VLANs which are used for GEM ports may either be drawn by the system in a default manner, or the block of VLANs used for GEM ports may be configured to pull from a defined contiguous block of VLANs.

Creating a bridge on a GEM port which already exists will not require an additional reserved VLAN.

Figure 125: The default uses VLANs from the top of the pool of usable VLANs



When the default setting is used the system draws VLANs from the top of the available pool of VLANs, starting with VLAN 4090, then decrements for each new GEM port as needed. The user must plan for the usage of the VLANs, so they do not use a higher range VLAN for normal bridged traffic.

The configured reserved VLAN block defines the range of reserved VLANs. When the range is depleted no more GEM ports will be allowed until the range is expanded. The reserved range is protected from use for creating bridge.



Note: There is not a protected range when using the default for GEM port VLANs. The system will decrement the VLAN for each GEM port that is created, starting at 4090. If there is a conflict between a GEM port VLAN and a VLAN assigned for data, whether for bridging, then traffic for the GEM port and the data VLAN will be affected.

Two ideas which need to be understood about reserved VLANs for GEM ports:

- The reserved VLAN block is reserved system wide (no other card can use those VLANs for bridges)
- A reserved VLAN only uses up a VLAN on that particular OLT port (whether reserved or by the default method)

Configuring the VLAN block

With the configurable VLAN block users need to plan for the number of VLANs which will be used for GEM ports (See [Planning for GEM ports, page 954](#)). Once the location and size of the VLAN block are set, the system will draw from the VLAN block from the lower VLAN and increment for each GEM port which is added. Unlike the default GPON GEM port VLANs, the configured VLAN block range is protected from using a VLAN range which is already user assigned or creating a VLAN which is in the protected range.

The location and size of the VLAN are defined by the **reservedVlanIdStart** and **reservedVlanIdCount** parameters in the system profile. When **reservedVlanIdCount** is 0 the default method is used for VLAN GEMs.

```
reservedVlanIdStart: --> {0}
reservedVlanIdCount: --> {0}
```

To define the VLAN block, update **system 0**; this example sets a VLAN block from VLAN 2000 to 2199:

```
zSH> update system 0

system 0
Please provide the following: [q]uit.
.....
(parameters deleted from example)
```

```

.....
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {NONE(0)}:
reservedVlanIdStart: --> {0}: 2000
reservedVlanIdCount: --> {0}: 200
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

If users attempt to configure a VLAN block where there is an existing VLAN, the system will block the attempt. If a bridge with VLAN 200 exists and users try to configure the reserved VLAN block from 190 to 209, an error will be displayed (in bold).

```
zSH> update system 0
```

```

system 0
Please provide the following: [q]uit.
syscontact: -----> {Zhone Global Services and
Support 7195 Oakport Street Oakland Ca. (877) Zhone20
(946-6320) Fax (510)777-7113 support@zhone.com}:
sysname: -----> {Zhone MxK}:
.....
(parameters deleted from example)
.....
secure: -----> {disabled}:
webinterface: -----> {enabled}:
options: -----> {NONE(0)}:
reservedVlanIdStart: --> {2000}: 190
reservedVlanIdCount: --> {200}: 20
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Cannot change reserved VLANs in system profile.
Bridge, Host, or IP already uses VLAN in desired reserved
VLAN block
Starting over....
syscontact: -----> {Zhone Global Services and
Support 7195 Oakport Street Oakland Ca. (877) Zhone20
(946-6320) Fax (510)777-7113 support@zhone.com}:

```

If users attempt to build a bridge interface with a VLAN from the configured block, the system will block the attempt. So if users create a VLAN block of reserved VLANs from 2000 to 2199, then try to create a bridge interface, an error will be displayed:

```

zSH> bridge add 1-1-6-5/eth downlink vlan 2100
Error: Cannot create bridge on interface "1-1-6-5/eth".
      vlanId is reserved, see system profile.

```



Caution: The reserved VLAN block is defined for all OLTs on the system and those VLANs are reserved for the whole system.

Planning for GEM ports

Configuring the reserved VLAN block for how many GEM ports takes planning. Depending on the number and type of zNIDs (ONTs) used there will be different number of GEM ports that will be used.

GEM ports use a reserved VLAN but just for that OLT port. If users have a 100 GEM ports added on one OLT port, it will not affect the number of GEM ports on another OLT port.

Table 101 shows an example of GEM port usage for three popular Zhone ONTs.



Note: When using Unified Service Provisioning, a GEM port will be internally created for the SNMP interface for USP. The internally created GEM port IDs start at 3829, and grow upward. There can be up to 64 of GEM ports since there are maximum 64 ONUs per OLT link. Each of those GEM ports consumes a Reserved VLAN.

Table 101: Number of GEM ports needed varies by ONT model and configuration

Model	Ports	Max GEM ports	Common # GEMs used	Calculations
	zNID 2427 4 GigE 2 POTS 4 SSID (wireless)	Under the condition of 1:1 mapping of VLANs to GEM ports in Unified Service Provisioning RG mode: 1 for POTS 4 for SSID/Data 1 for Multicast Video 1 for Video on Demand 1 for Remote Management Access and /or TR-069 1 for MXK Unified Service Provisioning 1 for MXK CPE-Manager 10 total	1 for POTS 1 for Video 1 for Data 1 for Remote Management Access and /or TR-069 1 for MXK Unified Service Provisioning 1 for MXK CPE-Manager 6 common	Max GEMs 10 x 32 devices per OLT = 320 VLANs need to be reserved 10 x 64 devices per OLT = 640 VLANs need to be reserved Common GEMS 6 x 32 devices per OLT = 192 VLANs need to be reserved 6 x 64 devices per OLT = 384 VLANs need to be reserved

Table 101: Number of GEM ports needed varies by ONT model and configuration

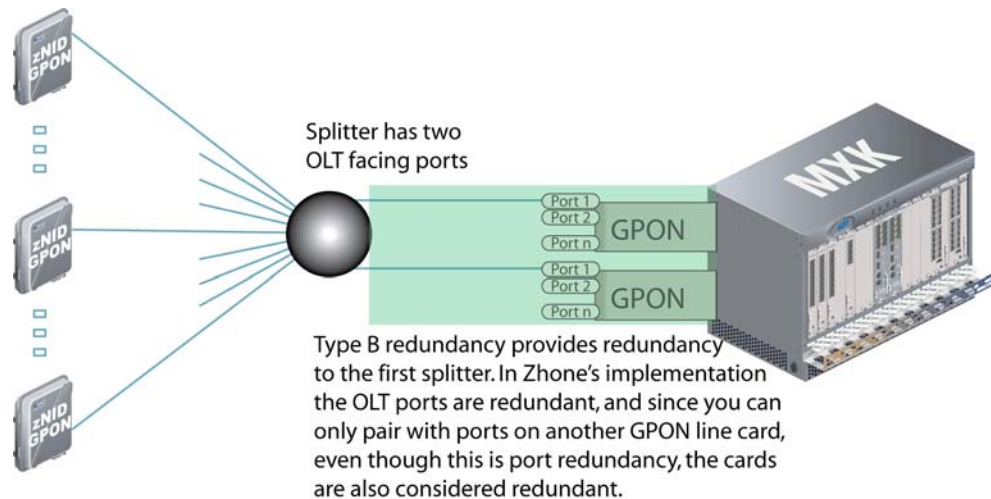
Model	Ports	Max GEM ports	Common # GEMs used	Calculations
	zNID 2520 4 FE 4 POTS	Under the condition of 1:1 or 1:many UNI to GEM port: 1 for POTS 4 for Data or Video 1 for Remote Management Access 6 total	1 for POTS 4 for Data or Video 1 for Remote Management Access 6 total	6 x 32 devices per OLT = 192 VLANs need to be reserved 6 x 64 devices per OLT = 384 VLANs need to be reserved
	zNID 8324 24 FE 24 POTS	Under the condition of 1:1 or 1:many UNI to GEM port: 24 for POTS 24 for data and video 48 total	2 POTS lines and 2 FE lines per subscriber. 2 GEMs per subscriber 24 total GEMs per ONT	24 x 32 devices = 768 VLANs need to be reserved 24 x 64 devices = 1536 VLANs need to be reserved

GPON type B redundancy

The MXK supports GPON type B redundancy as specified in the ITU-T G.984.1 standards specification.

Type B GPON redundancy doubles both the OLT ports on the GPON line card and the optical fiber between the OLT ports and the optical splitter which is closest to the OLTs. Users must use a splitter with two input/output ports on the OLT side. Outages on fiber from the OLT to the first splitter can be recovered from; With Type B redundancy there is no redundancy from the splitter to the ONT.

Figure 126: GPON type B redundancy



With Zhone's GPON type B redundancy, a GPON redundancy group can have two GPON OLT ports and the two GPON OLT ports must be on different GPON line cards. The ports can be on a 4 port or 8 port GPON line card. So even though the cards themselves are not redundant, their ports may be. Two 4 port GPON line cards can provide redundancy for a single 8 port GPON line card. Since it is port level redundancy and not card level redundancy, the port numbers on one card do not need to match the port number on the second card.

A single OLT port cannot be configured in two groups at the same time.

When the OLT ports in a GPON redundancy group are added, the active and standby port are based on whether they are added as a primary or secondary interface in the **line-red add** command. If users reboot the MXK system (or reboot both cards which have the redundant ports), the OLT port which comes up first and is able to pass traffic will be the active port.

In a redundancy group one OLT port is always assigned as active and the other standby. If an active OLT port fails, the standby takes over and becomes active. Note that OLT redundancy is non-revertive; that is, a previously active OLT port which has failed does not become active when the reason for the failover is resolved. The current active port will stay active until that port/line

fails, then the standby (if the initial issue was resolved) will once again become the active port.

When a standby port is added to the redundancy group and comes up, the card with the active port copies over the configuration database and routing tables to the standby OLT port on the second card. As configuration changes are made to the active port, the standby port is automatically updated.



Note: Create the line redundancy before building interfaces. If users add a port with existing interfaces as primary port of the redundant pair, users will need to perform a **slot reboot** on the GPON card with the secondary port after adding the redundancy.

Configuring GPON line redundancy

1 Show that there is no redundancy

Users can verify whether redundancy has been added by using the **gponolt show redund** command:

```
zSH> gponolt show redund
Redundancy --- Redundancy Peer ---
OLT      Interface      Status      State      OLT
Interface
=====
=====
3/1      1-3-1-0      OOS
3/2      1-3-2-0      OOS
3/3      1-3-3-0      OOS
3/4      1-3-4-0      UP

Redundancy --- Redundancy Peer ---
OLT      Interface      Status      State      OLT
Interface
=====
=====
4/1      1-4-1-0      OOS
4/2      1-4-2-0      OOS
4/3      1-4-3-0      OOS
4/4      1-4-4-0      UP
```

This display shows no redundancy because there is no information showing in the **Redundancy State** or the **OLT** and **Interface** columns.

Users can also show redundancy for a specific port using the **line-red show interface/physical interface** type command:

```
zSH> line-red show 1-3-4-0/gponolt
The 1-3-4-0/gponolt is not part of any redundancy
group
```

2 Add line redundancy

```
zSH> line-red add pri 1-3-4-0/gponolt sec 1-4-4-0/gponolt
```

This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] **yes**
Waiting for command completion.....

Protection pair has been created. 1-3-4-0/gponolt is primary and 1-4-4-0/gponolt is secondary.

3 Verify the line redundancy

 **Note:** You should wait until redundancy is confirmed before changing any provisioning on the port. Verify the redundancy using one of the following show commands before adding or deleting bridge interfaces on the OLT port.

Notice that the **gponolt show redund** command will warn users if an OLT port is not yet configured for redundancy.

```
zSH> gponolt show redund
ERROR! Failed to collect status for OLT 4/4
                                Redundancy ---
Redundancy Peer ---
OLT      Interface      Status      State      OLT
Interface
=====
=====
3/11-3-1-000S
3/21-3-2-000S
3/31-3-3-000S
3/41-3-4-0UPPrimary4/41-4-4-0

                                Redundancy ----
Redundancy Peer ----
OLT      Interface      Status      State      OLT
Interface
=====
=====
4/11-4-1-000S
4/21-4-2-000S
4/31-4-3-000S
```

When the OLT port is ready, it will be displayed in the gponolt show redund command:

```
zSH> gponolt show redund
                                Redundancy ---
Redundancy Peer ---
OLT      Interface      Status      State      OLT
Interface
=====
=====
3/11-3-1-000S
```

```

3/21-3-2-00OS
3/31-3-3-00OS
3/41-3-4-0UPPrimary4/41-4-4-0

```

```

Redundancy ----
Redundancy Peer ----
OLT      Interface      Status      State      OLT
Interface
=====
=====
4/11-4-1-00OS
4/21-4-2-00OS
4/31-4-3-00OS
4/41-4-4-0Trfc-DisableSecondary3/41-3-4-0

```

Users can also show the redundancy by the specific line

```

zSH> line-red show 1-3-4-0/gponolt
redundancy status for 1-3-4-0/gponolt:
      NOREBOOT standbytx ENABLE timeout 0
NONREVERTIVE revert timeout 0
Interface-Type      Interface-Name
Oper-State Oper-Status
=====
=====
Primary      1-3-4-0/gponolt      Active
UP
Secondary    1-4-4-0/gponolt      Standby
Trfc-Disable

```

4 Create interfaces on the GPON OLT port

Create a bridge interface:

```

zSH> bridge add 1-3-4-501/gponport gtp 1 downlink vlan
200 tagged
Adding bridge on 1-3-4-501/gponport
Created bridge-interface-record
1-3-4-501-gponport-200/bridge

```

Show the bridge:

```

zSH> bridge show
Orig
Type VLAN/SLAN VLAN/SLAN Physical Bridge St Table Data
-----
upl Tagged 200 1/a/5/0/eth ethernet5-200/
bridge UP S VLAN 200 default
dwn Tagged 200 1/3/4/1/gpononu 1-3-4-501-gponport-200/bridge UP D
00:00:86:43:3c:e4

```

5 Test line redundancy

a Bounce the port

```
zSH> port bounce 1-3-4-0/gponolt
1-3-4-0/gponolt set to admin state DOWN
1-3-4-0/gponolt set to admin state UP
```

b Show the line redundancy

```
zSH> line-red show 1-4-4-0
redundancy status for 1-4-4-0/gponolt:
      NOREBOOT standbytx ENABLE timeout 0
NONREVERTIVE revert timeout 0
Interface-Type      Interface-Name
Oper-State Oper-Status
=====
=====
Primary      1-3-4-0/gponolt      Standby
Trfc-Disable
Secondary    1-4-4-0/gponolt      Active
UP
```

c Show the bridge.

Notice that even though the port on card 4 is now the active port the name of the bridge does not change (and makes it look like the bridge is coming from the port on card 3, 1-3-4-501-gponport-200/bridge).

```
zSH> bridge show
      Orig
Type VLAN/SLAN VLAN/SLAN Physical      Bridge      St Table
Data
-----
-----
upl      Tagged 200 1/a/5/0/eth      ethernet5-200/
bridge UP S VLAN 200 default
dwn      Tagged 200 1/3/4/1/gpononu 1-3-4-501-gponport-200/bridge UP D
00:00:86:43:3c:e4
```

d Bounce the other port to get it to return to the initial redundancy state

```
zSH> port bounce 1-4-4-0
1-4-4-0 set to admin state DOWN
1-4-4-0 set to admin state UP
```

e Show the line redundancy

```
zSH> line-red show 1-4-4-0
redundancy status for 1-4-4-0/gponolt:
      NOREBOOT standbytx ENABLE timeout 0
NONREVERTIVE revert timeout 0
Interface-Type      Interface-Name
Oper-State Oper-Status
=====
=====
Primary      1-3-4-0/gponolt      Active
UP
```

```

Secondary      1-4-4-0/gponolt      Standby
Trfc-Disable

```

f Show the bridge

```

zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical      Bridge      St Table
Data
-----
-----
upl           Tagged 200 1/a/5/0/eth      ethernet5-200/
bridge UP S VLAN 200 default
dwn           Tagged 200 1/3/4/1/gpononu 1-3-4-501-gponport-200/bridge UP D
00:00:86:43:3c:e4

```

Removing a redundant OLT port

Redundancy may be removed from an OLT port, however there are limitations. The original primary port cannot be removed. Active ports can also not be removed.

To resolve downed ports which are on the primary port, resolve the problem with the port (whether downed link or card issue). Resolving the problem can include replacing the card with a new card, then running the **card add** command. When the new card comes up, the redundancy will be reestablished.

1 Show the current status of the redundancy group

```

zSH> line-red show 1-3-4-0/gponolt
redundancy status for 1-3-4-0/gponolt:
      NOREBOOT standbytx ENABLE timeout 0
      NONREVERTIVE revert timeout 0

Interface-Type      Interface-Name
Oper-State Oper-Status
=====
=====
Primary      1-3-4-0/gponolt      Active
UP
Secondary    1-4-4-0/gponolt      Standby
Trfc-Disable

```

2 Remove the standby port from the redundancy group

```

zSH> line-red remove 1-4-4-0/gponolt
Interface 1-4-4-0/gponolt is no longer in a protection
group.

```



Note: Notice that users cannot remove the primary port of a redundant pair even if it is in standby mode. Users also cannot remove the active port (even if it was initially the secondary port of a redundant port).

3 Show that the redundant port has been removed



Note: Users should wait until users confirm that redundancy has been removed before changing any provisioning on the port. Verify the redundancy using one of the following show commands before adding or deleting bridge interfaces on the OLT port.

```
zSH> line-red show 1-3-4-0/gponolt
redundancy status for 1-3-4-0/gponolt:
      NOREBOOT standbytx ENABLE timeout 0
NONREVERTIVE revert timeout 0
```

The 1-3-4-0/gponolt is not part of any redundancy group

Switchover between active and standby OLT ports

A switchover from active to standby OLT ports can be done automatically or forced manually using the **port bounce** command.

A switchover from standby to active OLT ports can be done manually by using the **line-red switch** command.

Automatically switched from active to standby

A switchover can be triggered automatically when:

- Loss of signal to all ONUs connected to the active OLT port occurs. This could be caused by:
 - The Fiber between the splitter and MXK is down (i.e. the fiber is cut or pulled)
 - Loss of all ONUs on this OLT port

If one or more ONUs go down with still a few ONUs active, it would not indicate a fiber failure between the splitter and MXK, and hence no action is taken by the SLMS software.
 - Loss or damage of splitter
- An SFP for this OLT port is damaged so it does not pass signal or the SFP is removed
- The GPON card is deleted or the card is rebooted
- The GPON card is physically pulled or removed from the chassis

When a switchover happens automatically, it raises an alarm.

Manually switched from active to standby

A manual switchover from active to standby OLT port can be done by operator by using the **port bounce interface/physical interface type** command on an active OLT port.

```
zSH> port bounce 1-3-4-0/gponolt
1-3-4-0/gponolt set to admin state DOWN
1-3-4-0/gponolt set to admin state UP
```

Manually switched from standby to active

A manual switchover from standby to active can be done by operator by using the **line-red switch interface/physical interface type** command on a standby port:

1) Show the Oper-State

```
zSH> line-red show 1-4-1-0/gponolt
> Interface-Type      Interface-Name      Oper-State Oper-Status
> =====
> Primary            1-4-1-0/gponolt    Active      UP
> Secondary          1-5-1-0/gponolt    Standby     Trfc-Disable
```

2) Switch the standby OLT port to active

```
zSH> line red switch 1-5-1-0/gponolt
1-5-1-0/gponolt is now active and 1-4-1-0/gponolt is
standby.
```

3) Check the Oper-State

```
zSH> line-red show 1-4-1-0/gponolt
Interface-Type      Interface-Name      Oper-State Oper-Status
> =====
> Primary            1-4-1-0/gponolt    Standby     Trfc-Disable
> Secondary          1-5-1-0/gponolt    Active      UP
```

GPON redundancy configuration limitations

The following limitations apply to GPON redundancy configurations:

- When a standby port is added, the configuration information is automatically inherited. If a port is configured as standby, the user cannot enter configuration on the port.
- The card which has proposed secondary ports must be a running card. Before users can use a newly installed card, users must add the card using the **card add** command.

- Users cannot add a secondary OLT port which has any added ONUs/ONTs, whether active or not. The port cannot be provisioned with logical interfaces, whether bridge or IP.

If there are active ONUs/ONTs in a standalone port that is being attempted to added as a standby to a redundancy group, the command is rejected. However, a OLT port with active ONUs/ONTs can be moved into a redundancy group as the primary active port.

- A OLT port may only be a member of one redundancy group.
- A OLT port may only be made redundant with another OLT port.
- The following rules apply to deleting ports from OLT redundancy groups:
 - An active port can never be deleted from the redundancy group. If the active port is the secondary port of the redundancy group, neither port can be removed.
 - Only the secondary port of a redundancy group can be deleted (and only when not active).
 - The primary port can never be deleted from the redundancy group.
- Upgrades cannot be scheduled on standby ports



Note: If a switchover event is triggered when an upgrade is in progress, the upgrade is re-queued for the ONUs/ONTs that were in progress as well as the ONUs/ONTs that were currently queued.

GPON extended reach

The MXK GPON solution supports extended reach. The maximum distance between the MXK and the farthest ONT/ONU is 60 Km. The requirement when deploying ONTs are the maximum distance between two ONTs cannot exceed 20Km.

A key point to keep in mind when deploying GPON is optical budget. This becomes more critical when the distance between MXK and ONT is higher. Listed below are factors that affect optical budget:

1. Higher the distance, higher the optical loss
2. Greater the number of splits, greater the optical loss
3. Cascading splitters causes higher loss than non-cascaded splits
4. Every connector introduces optical loss
5. Dirty fiber increases optical loss

It is recommended when deploying GPON:

1. Always clean the fiber before deploying
2. Use only fiber cleaning kit to clean fiber and not any other cleaning solution
3. Cap the unused splitters
4. Calculate the optical budget based on distance and fiber
5. Always plan for some margin in optical budget

Recommendations for extended reach

40 Km:

- Maximum Distance between MXK and farthest ONT: 40 Km
- Minimum Distance between MXK and closest ONT to MXK: 20 Km
- Maximum distance between any two ONTs: 20 Km (Note this is always a constant)

50 Km:

- Maximum Distance between MXK and farthest ONT: 50 Km
- Minimum Distance between MXK and closest ONT to MXK: 30 Km
- Maximum distance between any two ONTs: 20 Km (Note this is always a constant)

60 Km:

- Maximum Distance between MXK and farthest ONT: 60 Km

- Minimum Distance between MXK and closest ONT to MXK: 40 Km
- Maximum distance between any two ONTs: 20 Km (Note this is always a constant)

Command to measure the distance between MXK and ONT

To measure the approximate distance between MXK and ONU, use the **onu status** command. The distance field shows the measurement in km.

```
zSH> gpononu status 4/1/1
```

ID	Onu	OperStatus	ConfigState	Download State	OLT Rx Power	ONT Rx Power	Distance (KM)	GPON OnuStatus
1	1-4-1-1	Up	Active	NoUpgrade	-19.2 dBm	-20.0 dBm	18	Active

Commands to enable extended reach

By default, the MXK supports a maximum distance of 20 Km between MXK and ONT. The two parameters used to increase the supported distance on GPON are the *max-rt-propagation-delay* and *min-rt-propagation-delay* parameters in the **gpon-olt-config** profile. The default value for these parameters are 200 and 0 respectively. For every 1 Km increase the value of both these parameters have to be increased by a value of 10. So to increase the distance by 10 Km from the default value of 20 Km to 30 Km, change the *max-rt-propagation-delay* parameter in the **gpon-olt-config** profile to 300 and the *min-rt-propagation-delay* parameter in the **gpon-olt-config** profile to 100. Both these parameters have to be modified together as they go together in ensuring all ONTs can communicate with the MXK.

View the values of a profile with the get command:

```
zSH> get gpon-olt-config 1-1-1-0/gponolt
```

```
gpon-olt-config 1-1-1-0/gponolt
max-rt-propagation-delay: ----> {200}
max-onu-response-time: -----> {50}
preassigned-eqd: -----> {0}
los-alpha: -----> {4}
lof-alpha: -----> {4}
loam-alpha: -----> {3}
scrambler: -----> {enabled}
fec-mode: -----> {disabled}
auto-learn: -----> {enabled}
power-level: -----> {0}
guard-bit-count: -----> {32}
dba-mode: -----> {predictive}
gem-block-size: -----> {16}
us-ber-interval: -----> {5000}
ds-ber-interval: -----> {5000}
ber-sf-threshold: -----> {3}
ber-sd-threshold: -----> {5}
```

```

fec-request: -----> {disabled}
key-exchange: -----> {disabled}
min-rt-propagation-delay: ----> {0}
min-onu-response-time: -----> {10}
eqd-measure-cycles: -----> {5}
drift-ctrl-interval: -----> {1000}
drift-ctrl-limit: -----> {3}
alloc-cycle-length: -----> {2}
min-us-alloc: -----> {16}
ack-timeout: -----> {2000}
pls-max-alloc-size: -----> {120}
dba-cycle: -----> {2}
sr-dba-reporting-block-size: -> {48}

```

Update the profile with the update command:

```
zSH> update gpon-olt-config 1-1-1-0/gponolt
```

```

gpon-olt-config 1-1-1-0/gponolt
max-rt-propagation-delay: ----> {200} 300
max-onu-response-time: -----> {50}
preassigned-eqd: -----> {0}
los-alpha: -----> {4}
lof-alpha: -----> {4}
loam-alpha: -----> {3}
scrambler: -----> {enabled}
fec-mode: -----> {disabled}
auto-learn: -----> {enabled}
power-level: -----> {0}
guard-bit-count: -----> {32}
dba-mode: -----> {predictive}
gem-block-size: -----> {16}
us-ber-interval: -----> {5000}
ds-ber-interval: -----> {5000}
ber-sf-threshold: -----> {3}
ber-sd-threshold: -----> {5}
fec-request: -----> {disabled}
key-exchange: -----> {disabled}
min-rt-propagation-delay: ----> {0} 100
min-onu-response-time: -----> {10}
eqd-measure-cycles: -----> {5}
drift-ctrl-interval: -----> {1000}
drift-ctrl-limit: -----> {3}
alloc-cycle-length: -----> {2}
min-us-alloc: -----> {16}
ack-timeout: -----> {2000}
pls-max-alloc-size: -----> {120}
dba-cycle: -----> {2}
sr-dba-reporting-block-size: -> {48}
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

GPON Business Applications

Multicast VPN point-to-point service support on a wire bridge for GPON

MXK allows IP multicast through a wire bridge to support Multicast VPN (MVPN) point-to-point service for GPON. This allows multicast video conferencing equipment to communicate over a wire bridge. The MXK does not apply any IGMP snooping or proxy functions on this type of service. Management of multicast streams is performed by the equipment within the customer's VPN.

MVPN point-to-point service is only supported on wire bridge for GPON. It is not supported for TLS or other bridge types for GPON line types. All other line types support MVPN point-to-point over wire bridges or TLS bridges.

MVPN point-to-point is automatically added to wire bridges:

```
zSH> bridge add 1-4-1-502/gponport gtp 1 wire vlan 101
tagged
Adding bridge on 1-4-1-502/gponport
Created bridge-interface-record 1-4-1-502-gponport-101/
bridge
```

Upstream multicast video support

The upstream multicast video feature are supported on some OMCI-based zNIDs (e.g. zNID-GPON-2520) for applications like video surveillance.

The upstream multicast video feature can be turned on the OLT side by using the **update bridge-interface-record** command.

The following example turns on the upstream multicast video on a downlink bridge that was created on GEM port 901 for video services:

```
zSH> update bridge-interface-record
1-13-1-901-gponport-300/bridge
bridge-interface-record 1-13-1-901-gponport-300/bridge
Please provide the following: [q]uit.
vpi: -----> {0}:
vci: -----> {0}:
vlanId: -----> {500}:
stripAndInsert: -----> {false}:
customARP: -----> {false}:
filterBroadcast: -----> {false}:
learnIp: -----> {true}:
learnUnicast: -----> {true}:
maxUnicast: -----> {5}:
learnMulticast: -----> {true}:
forwardToUnicast: -----> {false}:
forwardToMulticast: -----> {false}:
forwardToDefault: -----> {true}:
```

```

bridgeIfCustomDHCP: -----> {false}:
bridgeIfIngressPacketRuleGroupIndex: -> {0}:
vlanIdCOS: -----> {0}:
outgoingCOSOption: -----> {disable}:
outgoingCOSValue: -----> {0}:
s-tagTPID: -----> {0x8100}:
s-tagId: -----> {0}:
s-tagStripAndInsert: -----> {true}:
s-tagOutgoingCOSOption: -----> {s-tagdisable}:
s-tagIdCOS: -----> {0}:
s-tagOutgoingCOSValue: -----> {0}:
mcastControlList: -----> {}:
maxVideoStreams: -----> {0}:
isPPPoA: -----> {false}:
floodUnknown: -----> {false}:
floodMulticast: -----> {false}: true
bridgeIfEgressPacketRuleGroupIndex: --> {0}:
bridgeIfTableBasedFilter: -----> {NONE(0)}:
bridgeIfDhcpLearn: -----> {NONE(0)}:
mvrVlan: -----> {0}:
vlan-xlate-from: -----> {0}:
slan-xlate-from: -----> {0}:
.....
Save changes? [s]ave, [c]hange or [q]uit:s

```

ONT Inventory Report

With the GPON ONT inventory query command, **onu inventory**, user can generate an inventory report of all the GPON ONTs connected with the MXK per system, per GPON card, per OLT port, or per an individual ONT. The inventory report shows ONU ID, ONU serial number, Vendor ID, Model ID, ONT version, and Software version for each GPON ONT.

Without specifying any slot ID or port ID, the **onu inventory** command shows all the GPON ONTs connected with this MXK:

```
zSH> onu inventory
Processing list of 960
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
```

ID	Interface	Serial Number	Vendor ID	Model ID	ONT Version	Software Version
1	1-9-1-1	0318E1BA	ZNTS	2424	S2.5.058	S2.5.058
2	1-9-1-2	09181341	ZNTS	-	-	-
3	1-9-1-3	09181320	ZNTS	2510A	00142-00011-C3	R3.4.2.263sbn
4	1-9-1-4	-	-	-	-	-
5	1-9-1-5	C0800012	GPAC	-	-	-
6	1-9-1-6	0318F730	ZNTS	2424	S2.5.058	S2.5.058
7	1-9-1-7	0317592B	ZNTS	2425	S2.5.039	S2.5.039
8	1-9-1-8	0318E154	ZNTS	-	-	-
9	1-9-1-9	0318E02D	ZNTS	-	-	-
10	1-9-1-10	0318E08B	ZNTS	-	-	-
11	1-9-1-11	0318E057	ZNTS	2424	S2.5.058	S2.5.058
12	1-9-1-12	0318F3EE	ZNTS	2424	S2.5.056	S2.5.056
13	1-9-1-13	-	-	-	-	-
14	1-9-1-14	-	-	-	-	-
15	1-9-1-15	-	-	-	-	-
16	1-9-1-16	-	-	-	-	-

<SPACE> for next page, <CR> for next line, A for all, Q to quit

The following example generates a report for all the GPON ONTs connected to the GPON card in the slot 9:

```
zSH> onu inventory 9
Processing list of 512
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
```

ID	Interface	Serial Number	Vendor ID	Model ID	ONT Version	Software Version
1	1-9-1-1	0318E1BA	ZNTS	2424	S2.5.058	S2.5.058
2	1-9-1-2	09181341	ZNTS	-	-	-
3	1-9-1-3	09181320	ZNTS	2510A	00142-00011-C3	R3.4.2.263sbn
4	1-9-1-4	-	-	-	-	-
5	1-9-1-5	C0800012	GPAC	-	-	-
6	1-9-1-6	0318F730	ZNTS	2424	S2.5.058	S2.5.058
7	1-9-1-7	0317592B	ZNTS	2425	S2.5.039	S2.5.039
8	1-9-1-8	0318E154	ZNTS	-	-	-

```

9   1-9-1-9   0318E02D  ZNTS      -        -        -
10  1-9-1-10  0318E08B  ZNTS      -        -        -
11  1-9-1-11  0318E057  ZNTS      2424     S2.5.058 S2.5.058
12  1-9-1-12  0318F3EE  ZNTS      2424     S2.5.056 S2.5.056
13  1-9-1-13      -        -        -        -        -
14  1-9-1-14      -        -        -        -        -
15  1-9-1-15      -        -        -        -        -
16  1-9-1-16      -        -        -        -        -
<SPACE> for next page, <CR> for next line, A for all, Q to quitq

```

The following example generates a report for all the GPON ONTs connected to the OLT port 9/1:

```

zSH> onu inventory 9/1
Processing list of 64
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes

```

ID	Interface	Serial Number	Vendor ID	Model ID	ONT Version	Software Version
1	1-9-1-1	0318E1BA	ZNTS	2424	S2.5.058	S2.5.058
2	1-9-1-2	09181341	ZNTS	-	-	-
3	1-9-1-3	09181320	ZNTS	2510A	00142-00011-C3	R3.4.2.263sbn
4	1-9-1-4	-	-	-	-	-
5	1-9-1-5	C0800012	GPAC	-	-	-
6	1-9-1-6	0318F730	ZNTS	2424	S2.5.058	S2.5.058
7	1-9-1-7	0317592B	ZNTS	2425	S2.5.039	S2.5.039
8	1-9-1-8	0318E154	ZNTS	-	-	-
9	1-9-1-9	0318E02D	ZNTS	-	-	-
10	1-9-1-10	0318E08B	ZNTS	-	-	-
11	1-9-1-11	0318E057	ZNTS	2424	S2.5.058	S2.5.058
12	1-9-1-12	0318F3EE	ZNTS	2424	S2.5.056	S2.5.056
13	1-9-1-13	-	-	-	-	-
14	1-9-1-14	-	-	-	-	-
15	1-9-1-15	-	-	-	-	-
16	1-9-1-16	-	-	-	-	-
17	1-9-1-17	-	-	-	-	-
18	1-9-1-18	-	-	-	-	-
19	1-9-1-19	-	-	-	-	-

```

<SPACE> for next page, <CR> for next line, A for all, Q to quitq

```

The following example generates a report for all the GPON ONT 9/1/1:

```

zSH> onu inventory 9/1/1

```

ID	Interface	Serial Number	Vendor ID	Model ID	ONT Version	Software Version
1	1-9-1-1	0318E1BA	ZNTS	2424	S2.5.058	S2.5.058

OMCI Statistics

The MXK obtains ONU statistics from the ONU using OMCI. The MXK sends standards based OMCI commands to retrieve statistics information. The statistics are maintained on the ONU in 15-minute intervals. There are 2 intervals of statistics that is stored in the ONU, current and previous. When an ONU is activated, the ONU starts storing statistics. These statistics are stored under the current category of statistics. After a 15 minute time period, the statistics value are reset. The statistics tracked during the past 15 minute period are stored as the previous interval. A new set of the current interval statistics is tracked. After every 15-minute period the current interval is saved as previous and a new current category is created with zeroed out values.

Display OMCI statistics for selected ONU(s) with the **gpononu statistics** command.

Syntax:

```
gpononu statistics [previous] [slot[/olt[/onu]] ifName]
```

previous

The system retrieves the statistics collected during the previous 15 minutes interval. Without **previous**, the system retrieves the statistics collected in current 15 minutes interval.

slot[/olt[/onu]]ifName

The ONU(s) users want to collect statistics on.

Example:

```
zSH> gpononu statistics previous 13/4/2
13/4/2 ONU Statistics (previous)
  Ethernet Performance Monitoring History Data - Port 1
    139 Interval Time
      0 Threshold Data Pointer
      0 FCS Errors
      0 Excessive Collision Counter
      0 Late Collision Counter
      0 Frame Too Long
      0 Buffer Overflows on Receive
      0 Buffer Overflows on Transmit
      0 Single Collision Frame Counter
      0 Multiple Collisions Frame Counter
      0 SQE Counter
      0 Deferred Transmission Counter
      0 Internal MAC Transmit Error Counter
      0 Carrier Sense Error Counter
      0 Alignment Error Counter
      0 Internal MAC Receive Error Counter
  Ethernet Performance Monitoring History Data 2 - Port 1
    no data available
  GEM Port Protocol Monitoring History Data - Port 1
    139 Interval Time
```



```

0 Threshold Data Pointer
0 Lost packets
0 Misinserted packets
0 Rx Packets
0 Rx Blocks
0 Tx Blocks
0 Impaired Blocks
GEM Port Protocol Monitoring History Data - Port 2
139 Interval Time
0 Threshold Data Pointer
0 Lost packets
0 Misinserted packets
0 Rx Packets
0 Rx Blocks
0 Tx Blocks
0 Impaired Blocks
GEM Port Protocol Monitoring History Data - Port 3
139 Interval Time
0 Threshold Data Pointer
0 Lost packets
0 Misinserted packets
0 Rx Packets
0 Rx Blocks
0 Tx Blocks
0 Impaired Blocks
GEM Port Protocol Monitoring History Data - Port 4
139 Interval Time
0 Threshold Data Pointer
0 Lost packets
0 Misinserted packets
0 Rx Packets
0 Rx Blocks
0 Tx Blocks
0 Impaired Blocks

```

[Table 102](#) defines the OMCI statistics displayed in the **gpononu statistics** command.

Table 102: OMCI statistics attributes

Attribute	Description
Interval end time	This attribute identifies the most recently finished 15-minute interval.
Threshold data pointer	This attribute points to an instance of the threshold data 1 and 2 managed entities that contains Performance Monitoring threshold values.
FCS errors	This attribute counts frames received on a particular interface that were an integral number of octets in length but failed the frame check sequence (FCS) check. The count is incremented when the MAC service returns the frameCheckError status to the link layer control (LLC) or other MAC user. Received frames for which multiple error conditions are obtained are counted according to the error status presented to the LLC.

Table 102: OMCI statistics attributes

Attribute	Description
Excessive collision counter	This attribute counts frames whose transmission failed due to excessive collisions.
Late collision counter	This attribute counts the number of times that a collision was detected later than 512 bit times into the transmission of a packet.
Frames too long	This attribute counts received frames that exceeded the maximum permitted frame size. The count is incremented when the MAC service returns the frameTooLong status to the LLC.
Buffer overflows on receive	This attribute counts the number of times that the receive buffer overflowed.
Buffer overflows on transmit	This attribute counts the number of times that the transmit buffer overflowed.
Single collision frame counter	This attribute counts successfully transmitted frames whose transmission was delayed by exactly one collision.
Multiple collisions frame counter	This attribute counts successfully transmitted frames whose transmission was delayed by more than one collision.
SQE counter	This attribute counts the number of times that the SQE test error message was generated by the PLS sublayer.
Deferred transmission counter	This attribute counts frames whose first transmission attempt was delayed because the medium was busy. The count does not include frames involved in collisions.
Internal MAC transmit error counter	This attribute counts frames whose transmission failed due to an internal MAC sublayer transmit error.

PON Statistics

This section includes the following topics:

- [View OLT statistics, page 975](#)
- [View ONU statistics, page 983](#)

PON statistics are the OLT or ONU statistics collected and reported by the MXK OLT.

The Downstream stats are the stats that were sent from the MXK to the ONU, and the upstream stats was sent from the ONU to the MXK.

View OLT statistics

The MXK OLT can report these stats types for an OLT interface: GPON physical layer stats for OLT (i.e. gpon), Ethernet layer stats (i.e. rmon), and interface layer stats (i.e. intf). The GPON physical layer stats are only available on OLT interfaces.

The MXK OLT can report these stats types for an ONU interface: ONU physical layer stats for ONU (i.e. onu) and interface layer stats (i.e. intf). The ONU physical layer stats are only available on ONU interfaces.

Collects and display OLT and ONU statistics with the **port statistics** command when specifying an OLT or ONU interface in the *inName/Type*.

Syntax:

```
port stats ifName/Type StatsType
```

ifName

interface name, in the format of *shelfID-SlotID-OLTID-ONUID*.

Type

interface type. e.g. gponolt, gpononu, eth, linegroup, etc.

To display stats for the OLT or ONU interface, users must use interface type **gponolt** or **gpononu**.

StatsType

statistics type. The possible stats types are:

- **intf**
refers to mib2 interface statistics. intf is the default value, if there is no stats type specified, system shows intf stats. It is valid for all interface type.
- **rmon**
refers to ethernet remote monitoring statistics. It is valid for ethernet or gponolt interfaces.
- **eth**

refers to ethernet dot3 statistics.

- **olt**

refers to GPON OLT traffic management statistics. It is valid for **gponolt** interfaces only.

- **onu**

refers to GPON ONU error statistics as reported by the MXK OLT. It is valid for **gpononu** interfaces only.

- **all**

refers to all statistics relevant to the interface type.

Viewing OLT stats

- 1 View the OLT interfaces. This step is optional.

```
zSH> list if-translate
```

```
Processing list of 896
```

```
if-translate 1-a-1-0/eth
if-translate ethernet1/linegroup
if-translate 1-a-5-0/ethproxy
if-translate 1-a-2-0/eth
if-translate ethernet2/linegroup
if-translate 1-a-3-0/eth
if-translate ethernet3/linegroup
if-translate 1-a-4-0/eth
if-translate 1-a-5-0/eth
if-translate ethernet5/linegroup
if-translate 1-a-6-0/ipobridge
if-translate ipobridge/linegroup
if-translate 1-a-1-0/rs232
if-translate 1-a-1-0/aal5proxy
if-translate 1-a-1-0-aal5proxy/linegroup
if-translate 1-a-2-0/dspproxy
if-translate 1-a-2-0-dspproxy/linegroup
if-translate 1-a-1-0-aal5proxy/atm
if-translate 1-a-1-0-aal5proxy/aal5
if-translate 1-a-1-0-aal5proxy/rfc1483
if-translate 1-a-2-0-dspproxy/atm
if-translate 1-a-2-0-dspproxy/aal5
if-translate 1-4-1-0/eth
if-translate 1-4-1-0-eth/linegroup
if-translate 1-4-2-0/eth
if-translate 1-4-2-0-eth/linegroup
if-translate 1-4-3-0/eth
if-translate 1-4-3-0-eth/linegroup
if-translate 1-4-4-0/eth
if-translate 1-4-4-0-eth/linegroup
if-translate 1-4-5-0/ethproxy
if-translate 1-4-5-0/eth
```

```

if-translate 1-4-5-0-eth/linegroup
if-translate 1-4-6-0/eth
if-translate 1-4-6-0-eth/linegroup
if-translate 1-4-7-0/eth
if-translate 1-4-7-0-eth/linegroup
if-translate 1-4-8-0/eth
...
if-translate 1-7-3-0/gponolt
...
if-translate 1-7-3-1/gpononu
...
896 entries found.

```

- 2 When specifying **all** as the stats type, the rmon, OLT and interface stats are displayed for this OLT interface.

```

zSH> port stats 1-4-4-0/gponolt all
***** rmon *****
Total Dropped Events                                0
Total Dropped Frames                                0
Total Bytes                                           0
Total Packets                                         0
Transmitted Packets                                  0
Received Packets                                     0
Transmitted Multicast Bytes                           0
Received Multicast Bytes                             0
Received Multicast Dropped Bytes                     0
Transmitted Average Throughput                       0
Received Average Throughput                         0
Transmitted Bandwidth Occupancy                     0
Received Bandwidth Occupancy                       0
Total Broadcast Packets                             0
Total Multicast Packets                             0
CRC Align Errors                                     0
Undersize Packets                                    0
Oversize Packets                                     0
Transmitted Oversize Packets                         0
Received Oversize Packets                           0
Fragments                                            0
Jabbers                                              0
Collisions                                           0
Transmitted No Errors                               0
Received No Errors                                  0
IPMC Bridged Packets                                0
IPMC Routed Packets                                 0
Transmitted IPMC Dropped Packets                    0
Received IPMC Dropped Packets                      0
Total Packets 0 to 64 Bytes                          0
Total Packets 65 to 127 Bytes                       0
Total Packets 128 to 255 Bytes                      0
Total Packets 256 to 511 Bytes                      0
Total Packets 512 to 1023 Bytes                     0
Total Packets 1024 to 1518 Bytes                    0
Total Packets 1519 to 2047 Bytes                    0

```

Total Packets 2048 to 4095 Bytes	0
Total Packets 4095 to 9216 Bytes	0
Received Packets 0 to 64 Bytes	0
Received Packets 65 to 127 Bytes	0
Received Packets 128 to 255 Bytes	0
Received Packets 256 to 511 Bytes	0
Received Packets 512 to 1023 Bytes	0
Received Packets 1024 to 1518 Bytes	0
Received Packets 1519 to 2047 Bytes	0
Received Packets 2048 to 4095 Bytes	0
Received Packets 4095 to 9216 Bytes	0
Transmitted Packets 0 to 64 Bytes	0
Transmitted Packets 65 to 127 Bytes	0
Transmitted Packets 128 to 255 Bytes	0
Transmitted Packets 256 to 511 Bytes	0
Transmitted Packets 512 to 1023 Bytes	0
Transmitted Packets 1024 to 1518 Bytes	0
Transmitted Packets 1519 to 2047 Bytes	0
Transmitted Packets 2048 to 4095 Bytes	0
Transmitted Packets 4095 to 9216 Bytes	0
***** olt *****	
Upstream Valid Gem Frames	0
Upstream Discarded Frames	0
Upstream Gem Frames	0
Upstream Omci Frames	0
Upstream Ploam Frames	0
Upstream Idle Ploam Frames	0
Downstream Valid Gem Frames	0
Downstream Discarded Frames	0
Downstream Gem Frames	0
Downstream Omci Frames	0
Downstream Ploam Frames	0
Downstream Idle Ploam Frames	0
Downstream Pon Valid Ethernet Packtes	0
Downstream Pon Cpu Packets	0
Downstream Transmitted Bytes	0
Upstream Pon Valid Packets	0
Upstream Pon Valid Not Idle Ploams	0
Upstream Pon Error Ploams	0
Upstream Pon Invalid Packets	0
Upstream Dropped Packets	0
Upstream Dropped Ploams Fifo Full	0
Downstream TM Valid Packets	0
Downstream TM Crc Packets	0
Downstream TM Dropped Cpu Packets	0
Downstream TM MAC Lookup Miss	0
Downstream TM Packets Forwarded From Hm To Pon	0
Downstream TM Packets Dropped Gem Pid Not Enabled	0
Downstream TM Q0 Valid Packets	0
Downstream TM Q0 Dropped Packets	0
Downstream TM Q1 Valid Packets	0
Downstream TM Q1 Dropped Packets	0
Downstream TM Q2 Valid Packets	0
Downstream TM Q2 Dropped Packets	0

```

Downstream TM Q3 Valid Packets 0
Downstream TM Q3 Dropped Packets 0
Downstream TM Q4 Valid Packets 0
Downstream TM Q4 Dropped Packets 0
Downstream TM Q5 Valid Packets 0
Downstream TM Q5 Dropped Packets 0
Downstream TM Q6 Valid Packets 0
Downstream TM Q6 Dropped Packets 0
Downstream TM Q7 Valid Packets 0
Downstream TM Q7 Dropped Packets 0
Upstream TM Dropped Cpu Packets 0
Upstream TM Dropped Packets Crc Error 0
Upstream TM Dropped Packets Security 0
Upstream TM Learn Failures 0
Upstream TM Q0 Valid Packets 0
Upstream TM Q0 Dropped Packets 0
Upstream TM Q1 Valid Packets 0
Upstream TM Q1 Dropped Packets 0
Upstream TM Q2 Valid Packets 0
Upstream TM Q2 Dropped Packets 0
Upstream TM Q3 Valid Packets 0
Upstream TM Q3 Dropped Packets 0
Upstream TM Q4 Valid Packets 0
Upstream TM Q4 Dropped Packets 0
Upstream TM Q5 Valid Packets 0
Upstream TM Q5 Dropped Packets 0
Upstream TM Q6 Valid Packets 0
Upstream TM Q6 Dropped Packets 0
Upstream TM Q7 Valid Packets 0
Upstream TM Q7 Dropped Packets 0
***** intf *****
Interface Name 1-4-4-0
Operational Status Down
Received Bytes 0
Received Packets 0
Received Multicast Packets 0
Received Broadcast Packets 0
Transmitted Bytes 0
Transmitted Unicast Packets 0
Transmitted Multicast Packets 0
Transmitted Broadcast Packets 0
Received Discards 0
Received Errors 0
Received Unknown Protocols 0
Transmitted Discards 0
Transmitted Errors 0
Speed Bits per Second *** n/a ***
Speed Megabits per Second 2400

```

- 3** When specifying **olt** as the stats type, only the GPON OLT physical layer statistics are displayed for this OLT interface.

```

zSH> port stats 1-7-3-0/gponolt olt
Upstream Valid Gem Frames 1390778452

```

Upstream Discarded Frames	0
Upstream Gem Frames	1390766390
Upstream Omci Frames	12062
Upstream Ploam Frames	2773259552
Upstream Idle Ploam Frames	2772149075
Downstream Valid Gem Frames	1408605291
Downstream Discarded Frames	3416
Downstream Gem Frames	1408595361
Downstream Omci Frames	9930
Downstream Ploam Frames	117890
Downstream Idle Ploam Frames	0
Downstream Pon Valid Ethernet Packtes	1408591816
Downstream Pon Cpu Packets	9930
Downstream Transmitted Bytes	2050960928301
Upstream Pon Valid Packets	1390766377
Upstream Pon Valid Not Idle Ploams	1110477
Upstream Pon Error Ploams	23
Upstream Pon Invalid Packets	0
Upstream Dropped Packets Inactive Ports	0
Upstream Dropped Ploams Fifo Full	0
Downstream TM Valid Packets	1408605259
Downstream TM Crc Packets	0
Downstream TM Dropped Cpu Packets	0
Downstream TM MAC Lookup Miss	0
Downstream TM Packets Forwarded From Hm To Pon	1005110436
Downstream TM Packets Dropped Gem Pid Not Enabled	3416
Downstream TM Q0 Valid Packets	1408595361
Downstream TM Q0 Dropped Packets	0
Downstream TM Q1 Valid Packets	0
Downstream TM Q1 Dropped Packets	0
Downstream TM Q2 Valid Packets	0
Downstream TM Q2 Dropped Packets	0
Downstream TM Q3 Valid Packets	0
Downstream TM Q3 Dropped Packets	0
Downstream TM Q4 Valid Packets	0
Downstream TM Q4 Dropped Packets	0
Downstream TM Q5 Valid Packets	0
Downstream TM Q5 Dropped Packets	0
Downstream TM Q6 Valid Packets	0
Downstream TM Q6 Dropped Packets	0
Downstream TM Q7 Valid Packets	0
Downstream TM Q7 Dropped Packets	0
Upstream TM Dropped Cpu Packets	0
Upstream TM Dropped Packets Crc Error	0
Upstream TM Dropped Packets Security	0
Upstream TM Learn Failures	0
Upstream TM Q0 Valid Packets	1390766390
Upstream TM Q0 Dropped Packets	0
Upstream TM Q1 Valid Packets	0
Upstream TM Q1 Dropped Packets	0
Upstream TM Q2 Valid Packets	0
Upstream TM Q2 Dropped Packets	0
Upstream TM Q3 Valid Packets	0
Upstream TM Q3 Dropped Packets	0

Upstream TM Q4 Valid Packets	0
Upstream TM Q4 Dropped Packets	0
Upstream TM Q5 Valid Packets	0
Upstream TM Q5 Dropped Packets	0
Upstream TM Q6 Valid Packets	0
Upstream TM Q6 Dropped Packets	0
Upstream TM Q7 Valid Packets	0
Upstream TM Q7 Dropped Packets	0

Table 103 defines the GPON OLT physical layer statistics displayed in the **port stats** *ifName/gponolt* command.

Note that the Downstream stats are the stats that were sent from MXK to ONU, and the upstream stats was sent from ONU to MXK.

Table 103: GPON OLT physical layer statistics attributes

Attribute	Description
Upstream Valid Gem Frames	The number of valid GEM frames sent in upstream direction.
Upstream Discarded Frames	Total number of discarded GEM frames sent in upstream direction.
Upstream Gem Frames	The number of GEM frames sent in the upstream direction.
Upstream Omci Framesr	The number of OMCI frames sent in the upstream direction.
Upstream Ploam Frames	Total number of Physical Layer Operations, Administration and Maintenance (PLOAM) frames sent in the upstream direction. This includes: • Total number of PLOAM messages, including idle PLOAMs. • Total number of valid PLOAM messages (not including idle PLOAMs) • Total number of PLOAM messages dropped due to Cyclic Redundancy Check (CRC) errors.
Upstream Idle Ploam Frames	Total number of idle PLOAM frames sent in upstream direction.
Downstream Valid Gem Frames	Total number of valid GEM frames sent in downstream direction.
Downstream Discarded Frames	The number of downstream packets discarded due to CRC errors, MAC lookup miss, congestion, etc.
Downstream Gem Frames	Total number of GEM frames sent in downstream direction.
Downstream Omci Frames	Total number of OMCI frames sent in downstream direction.
Downstream Ploam Frames	Total number of PLOAM frames sent in downstream direction.

Table 103: GPON OLT physical layer statistics attributes

Attribute	Description
Downstream Idle Ploam Frames	Total number of idle PLOAM frames sent in downstream direction.
Downstream Pon Valid Ethernet Packets	Total number of valid Ethernet packets sent in downstream direction.
Downstream Pon Cpu Packets	The number of downstream packets generated by the CPU (MIPS).
Downstream Transmitted Bytes	Total number of bytes transmitted sent in downstream direction.
Upstream Pon Valid Packets	Total number of valid PON packets sent in upstream direction.
Upstream Pon Valid Not Idle Ploams	Total number of valid non-idle PLOAM messages sent in upstream direction.
Upstream Pon Error Ploams	Total number of PON error PLOAM messages sent in upstream direction.
Upstream Pon Invalid Packets	The number of upstream errored packets.
Upstream Dropped Packets Inactive Ports	Total number of upstream packets that were dropped because the GEM port ID was not configured.
Upstream Dropped Ploams Fifo Full	Total number of upstream PLOAMs that were dropped because the FIFO buffer was full.
Downstream TM Valid Packets	Total number of valid packets that were sent in downstream direction.
Downstream TM Crc Packets	The number of dropped downstream packets due to CRC errors.
Downstream TM Dropped Cpu Packets	The number of dropped downstream packets originated by the CPU (MIPS).
Downstream TM MAC Lookup Miss	The number of downstream MAC lookup miss events.
Downstream TM Packets Forwarded From Hm To Pon	The number of downstream packets forwarded from the header modification stage to the PON.

Table 103: GPON OLT physical layer statistics attributes

Attribute	Description
Downstream TM Packets Dropped Gem Pid Not Enabled	The number of downstream packets dropped because the GEM port ID was not configured correctly.
Downstream TM QN Valid Packets (N=0 to 7)	The number of downstream packets forwarded by egress priority queue [0 to 7] to the PON. Queue 0 is the highest priority; queue 7 is the lowest priority. Packets in the lowest priority queues are dropped first. When the PON link is not active, this counter will not increment.
Downstream TM QN Dropped Packets (N=0 to 7)	The number of downstream packets dropped by egress priority queue [0 to 7] due to congestion. Queue 0 is the highest priority; queue 7 is the lowest priority. Packets in the lowest priority queues are dropped first. When the PON link is not active, this counter will not increment.
Upstream TM Dropped Cpu Packets	The number of upstream packets dropped by the CPU(MIPS), not sent to SGMI interface.
Upstream TM Dropped Packets Crc Error	The number of upstream packets that were dropped because of CRC errors.
Upstream TM Dropped Packets Security	Total number of upstream packets that were dropped because they didn't pass the security rules.
Upstream TM Learn Failures	MAC address learning failures from traffic sent in upstream direction that were due to a full FIFO buffer.
Upstream TM QN Valid Packets (N=0 to 7)	Number of upstream packets forwarded by egress priority queue [0 to 7] to the MxK. Queue 0 is the highest priority; queue 7 is the lowest priority. Packets in the lowest priority queues are dropped first. When the PON link is not active, this counter will not increment.
Upstream TM QN Dropped Packets (N=0 to 7)	Number of upstream packets dropped by egress priority queue [0 to 7] due to congestion. Queue 0 is the highest priority; queue 7 is the lowest priority. Packets in the lowest priority queues are dropped first. When the PON link is not active, this counter will not increment.

View ONU statistics

Viewing ONU stats

- 1 View the ONU interfaces. This step is optional.

```
zSH> list if-translate
```

```
Processing list of 896
```

```
if-translate 1-a-1-0/eth
```

```

if-translate ethernet1/linegroup
if-translate 1-a-5-0/ethproxy
if-translate 1-a-2-0/eth
if-translate ethernet2/linegroup
if-translate 1-a-3-0/eth
if-translate ethernet3/linegroup
if-translate 1-a-4-0/eth
if-translate 1-a-5-0/eth
if-translate ethernet5/linegroup
if-translate 1-a-6-0/ipobridge
if-translate ipobridge/linegroup
if-translate 1-a-1-0/rs232
if-translate 1-a-1-0/aal5proxy
if-translate 1-a-1-0-aal5proxy/linegroup
if-translate 1-a-2-0/dspproxy
if-translate 1-a-2-0-dspproxy/linegroup
if-translate 1-a-1-0-aal5proxy/atm
if-translate 1-a-1-0-aal5proxy/aal5
if-translate 1-a-1-0-aal5proxy/rfc1483
if-translate 1-a-2-0-dspproxy/atm
if-translate 1-a-2-0-dspproxy/aal5
if-translate 1-4-1-0/eth
if-translate 1-4-1-0-eth/linegroup
if-translate 1-4-2-0/eth
if-translate 1-4-2-0-eth/linegroup
if-translate 1-4-3-0/eth
if-translate 1-4-3-0-eth/linegroup
if-translate 1-4-4-0/eth
if-translate 1-4-4-0-eth/linegroup
if-translate 1-4-5-0/ethproxy
if-translate 1-4-5-0/eth
if-translate 1-4-5-0-eth/linegroup
if-translate 1-4-6-0/eth
if-translate 1-4-6-0-eth/linegroup
if-translate 1-4-7-0/eth
if-translate 1-4-7-0-eth/linegroup
if-translate 1-4-8-0/eth
...
if-translate 1-7-3-0/gponolt
...
if-translate 1-7-3-1/gpononu
...
896 entries found.

```

- 2** When specifying **onu** as the stats type, the ONU physical layer statistics are displayed for this ONU interface.

```

zSH> port stats 1-7-3-3/gpononu onu
Upstream BIP8 Errors                                0
Upstream FEC Corrected Bytes                        0
Upstream FEC Corrected Code-words                  0
Upstream FEC Uncorrectable Code-words               0
Upstream Received Code-words                       0
Upstream Unreceived Bursts                         0

```

```

Downstream Remote BIP8 Errors          0
Upstream Remote BIP8 Errors            0
Drift Of Window Indications            0
Message Error Message                  0

```

- 3** When specifying **all** as the stats type, only ONU stats type is available and displayed for this ONU interface.

```

zSH> port stats 1-7-3-3/gpononu all
*****  onu  *****

```

```

Upstream BIP8 Errors          0
Upstream FEC Corrected Bytes  0
Upstream FEC Corrected Code-words  0
Upstream FEC Uncorrectable Code-words  0
Upstream Received Code-words  0
Upstream Unreceived Bursts    0
Downstream Remote BIP8 Errors  0
Upstream Remote BIP8 Errors    0
Drift Of Window Indications    0
Message Error Message          0

```

[Table 104](#) defines the GPON ONU physical layer statistics displayed in the **port stats ifName/gpononu** command.

Table 104: GPON ONU physical layer statistics attributes

Attribute	Description
Upstream BIP8 Errors	Total number of upstream Bit-Interleaved Parity with eight bit (BIP8) errors per ONU-ID.
Upstream FEC Corrected Bytes	Total number of upstream FEC corrected bytes per ONU-ID.
Upstream FEC Corrected Code-words	Total number of upstream FEC corrected code words per ONU-ID.
Upstream FEC Uncorrectable Code-words	Total number of upstream FEC uncorrected code words per ONU-ID.
Upstream Received Code-words	Total number of upstream code words transmitted per ONU-ID.
Upstream Unreceived Bursts	Total number of upstream un-received bursts per ONU-ID.
Downstream Remote BIP8 Errors	Total number of downstream remote BIP8 errors per ONU-ID.
Upstream Remote BIP8 Errors	Total number of upstream remote BIP8 errors per ONU-ID.

Table 104: GPON ONU physical layer statistics attributes

Attribute	Description
Drift Of Window Indications	The number of times the average drift for the ONU exceeds the drift threshold.
Message Error Message	The number of error messages sent from the ONU.

GPON Alarms and Traps

This sections describes the following topics:

- [GPON Alarms, page 987](#)
- [GPON Traps, page 1010](#)

GPON Alarms

- [Monitor GPON alarms, page 987](#)
- [GPON BIP Threshold Crossing Monitor Alarms, page 987](#)
- [GPON High and Low Receive Power Threshold Alarms, page 992](#)
- [Rogue ONU detection and rogue ONU alarms, page 995](#)
- [ONU Dying Gasp Alarms, page 1007](#)
- [ONU Manual Reboot Alarms, page 1008](#)

Monitor GPON alarms

Users can monitor GPON alarms in different levels:

- If users want to view the standard GPON MAC alarms that generated on the ONU and detected on the OLT, use the **gpononu status** command. For example: LOS (Lost Of Signal) alarm or DG (Dying Gasp) alarm of an ONU.

For the details about the **gpononu status** command, refer to [Monitoring ONU Status and Alarms, page 910](#).

- If users want to view the internal alarms that generated on the ONU UNI ports (also called LAN facing ports or subscriber facing ports) and detected on the ONU, use the **gpononu alarms** command. For example: LanLos alarm of an Ethernet UNI port.

For the details about the **gpononu alarms** command, refer to [Monitoring ONU UNI ports Status and Alarms, Configuring ONU UNI port Admin Status and Port speed, page 913](#).

- If users want to view the ONU and OLT alarms that generated on the MXK system, use the **alarm show** command. For example: GPON BIP threshold crossing monitor alarms, GPON high and low receive power threshold alarms, or rogue ONU detection and rogue ONU alarms.

The GPON alarms reported with the **alarm show** command are described in the following section.

GPON BIP Threshold Crossing Monitor Alarms

Users can monitor BIP threshold crossing alarms, set the threshold for BIP errors on GPON, and also configure whether or not to auto-disable the ONU if

the threshold has been exceeded. BIP is a counter representing bit errors on the PON link to a specific ONU. This is configured on a per-OLT basis, but is monitored per ONU. To configure the GPON BIP threshold on all ONUs under an OLT, use the **update gpon-olt-config** command.

ONU raises a “bip threshold exceeded” alarm if bip-error-monitoring-mode in the gpon-olt-config profile is set to either monitorOnly or blockOnError and the alarm condition exists. When the alarm is set, the MXK will periodically restart the BIP error measurement. If the condition that cause the alarm is improved, a deactivated ONU is reactivated, and the alarm is cleared. The default interval for the periodic measurement is 5 minutes.

```
MXK-13> update gpon-olt-config 1-1-1-0/gponolt
gpon-olt-config 1-1-1-0/gponolt
Please provide the following: [q]uit.
max-rt-propagation-delay: ----> {200}:
max-onu-response-time: -----> {50}:
preassigned-eqd: -----> {0}:
los-alpha: -----> {4}:
lof-alpha: -----> {4}:
loam-alpha: -----> {3}:
scrambler: -----> {enabled}:
fec-mode: -----> {disabled}:
auto-learn: -----> {enabled}:
power-level: -----> {0}:
guard-bit-count: -----> {32}:
dba-mode: -----> {predictive}:
gem-block-size: -----> {16}:
us-ber-interval: -----> {5000}:
ds-ber-interval: -----> {5000}:
ber-sf-threshold: -----> {3}:
ber-sd-threshold: -----> {5}:
fec-request: -----> {disabled}:
key-exchange: -----> {disabled}:
min-rt-propagation-delay: ----> {0}:
min-onu-response-time: -----> {10}:
eqd-measure-cycles: -----> {5}:
drift-ctrl-interval: -----> {1000}:
drift-ctrl-limit: -----> {3}:
alloc-cycle-length: -----> {2}:
min-us-alloc: -----> {16}:
ack-timeout: -----> {2000}:
pls-max-alloc-size: -----> {120}:
dba-cycle: -----> {2}:
sr-dba-reporting-block-size: -> {48}:
protection-switchover-timer: -> {500}:
preamble-override: -----> {disabled}:
preamble-type-0: -----> {0x00}:
preamble-type-1: -----> {0x00}:
preamble-type-3-pre-range: ---> {0x0b}:
preamble-type-3-post-range: --> {0x08}:
preamble-type-3-pattern: -----> {0xaa}:
bip-error-monitoring-mode: ---> {monitorOnly}:
errors-per-sample-threshold: -> {100}:
```



```

errored-samples-threshold: ---> {10}:
bip-max-sample-gap: -----> {10}:
.....
Save changes? [s]ave, [c]hange or [q]uit:q

```

Table 105: BIP error threshold attributes in gpon-olt-config profile

Attribute	Description
bip-error-monitoring-mode	Disable or enable the BIP error monitoring. Values: disabled The BIP error monitoring feature is disabled. monitorOnly Monitor BIP errors. When the ONU crosses the BIP error threshold, trigger a local alarm and send a trap to ZMS. blockOnError Monitor BIP errors. When the ONU crosses the BIP error threshold, trigger a local alarm, send a trap to ZMS, disable ONU ranging and set ONU line status to DSA (i.e. disabled). Default: monitorOnly
errors-per-sample-threshold	If the number of BIP errors per sample exceeds this threshold, it is counted as an errored sample. Default: 100
errored-samples-threshold	If the number of errored samples exceed this sample threshold, report and disable the onu if in blockOnError mode, otherwise simply report the threshold as being exceeded. Default: 10
bip-max-sample-gap	If two adjacent errored samples were taken farther apart than this threshold, do not count the earlier sample as an errored sample. This value is in the unit of seconds. Default: 10

Configuring GPON BIP error threshold crossing monitor alarms

1 View the ONU status.

```
MXK-13> onu status 1/1/2
```

ID	Onu	OperStatus	Omci ConfigState	Gpon OnuStatus	Download State	OLT Rx Power	ONT Rx Power	Distance (KM)
2	1-1-1-2	Up	Active	Active	NoUpgrade	-23.8 dBm	-23.0 dBm	18

2 Configure the BIP error monitoring mode and thresholds as desired. This example changes the monitoring mode to **blockonerror**, and changes the BIP error threshold values.

```

MXK-13> update gpon-olt-config 1-1-1-0/gponolt
gpon-olt-config 1-1-1-0/gponolt
Please provide the following: [q]uit.
max-rt-propagation-delay: ----> {200}:
max-onu-response-time: -----> {50}:
preassigned-eqd: -----> {0}:

```

```

los-alpha: -----> {4}:
lof-alpha: -----> {4}:
loam-alpha: -----> {3}:
scrambler: -----> {enabled}:
fec-mode: -----> {disabled}:
auto-learn: -----> {enabled}:
power-level: -----> {0}:
guard-bit-count: -----> {32}:
dba-mode: -----> {predictive}:
gem-block-size: -----> {16}:
us-ber-interval: -----> {5000}:
ds-ber-interval: -----> {5000}:
ber-sf-threshold: -----> {3}:
ber-sd-threshold: -----> {5}:
fec-request: -----> {disabled}:
key-exchange: -----> {disabled}:
min-rt-propagation-delay: ----> {0}:
min-onu-response-time: -----> {10}:
eqd-measure-cycles: -----> {5}:
drift-ctrl-interval: -----> {1000}:
drift-ctrl-limit: -----> {3}:
alloc-cycle-length: -----> {2}:
min-us-alloc: -----> {16}:
ack-timeout: -----> {2000}:
pls-max-alloc-size: -----> {120}:
dba-cycle: -----> {2}:
sr-dba-reporting-block-size: -> {48}:
protection-switchover-timer: -> {500}:
preamble-override: -----> {disabled}:
preamble-type-0: -----> {0x00}:
preamble-type-1: -----> {0x00}:
preamble-type-3-pre-range: ---> {0x0b}:
preamble-type-3-post-range: --> {0x08}:
preamble-type-3-pattern: -----> {0xaa}:
bip-error-monitoring-mode: ---> {monitorOnly}:blockonerror
errors-per-sample-threshold: -> {100}: 99
errored-samples-threshold: ---> {10}:9
bip-max-sample-gap: -----> {10}:9
.....
Save changes? [s]ave, [c]hange or [q]uit:s

```

- 3 View the ONU status. This example assumes the BIP error on this ONU exceeded the threshold values. With the **blocknoerror** mode, the ONU will raise an alarm and be auto-disabled. The **GponOnuStatus** in this example shows a brief description about this ONU is inactive and EXCBIPDSA (i.e. exceeded BIP threshold, and ONU is disabled.).

```

MXK-13> onu status 1/1/2

```

ONT		Distance	Omci	Gpon	Download	OLT
ID	Onu	OperStatus	ConfigState	OnuStatus	State	Rx Power Rx Power
		(KM)				
==	=====	=====	=====	=====	=====	=====
=====	=====					

```

2  1-1-1-2  Down  Inactive  Inactive+EXCBIPDSA  None  error  error
0.0

```

GponOnuStatus acronym definitions:

"Active" - ONU is Active

"Inactive" - ONU is Inactive

"LOS" - Loss of Signal

"LOF" - Loss of Frame

"DOW" - Drift Of Window

"SF" - Signal Fail

"SD" - Signal Degrade

"LCDG" - Loss of GEM channel delineation

"RD" - Remote defect

"TF" - Transmitter Failure

"SUF" - Start-up Failure

"LOA" - Loss of Acknowledge

"DG" - Receive Dying-gasp

"OAML" - PLOAM Cell Loss

"MEM" - Message Error Message

"PEE" - Physical Equipment Error

"EXCBIPDSA" - Disable Onu, excessive BIP errors

"EXCBIP" - Excessive BIP errors. ONU is not disabled

"RXPWRDSA" - Upstream Rx Power out of range. ONU is disabled.

"RXPWRNOTDSA" - Upstream Rx Power out of range. ONU is not disabled.

4 View the raised alarms on this ONU at the system level.

```
MXK-13> alarm show
```

```

*****      Central Alarm Manager      *****
      ActiveAlarmCurrentCount           :8
      AlarmTotalCount                   :15
      ClearAlarmTotalCount              :7
      OverflowAlarmTableCount           :0

ResourceId      AlarmType      AlarmSeverity
-----
1-a-3-0/eth      linkDown      critical
1-1-1-0/gponolt  linkDown      critical
1-1-2-0/gponolt  linkDown      critical
1-1-3-0/gponolt  linkDown      critical
1-1-4-0/gponolt  linkDown      critical
1-1-1-2/gpononu  linkDown      minor
system           not_in_redundant_mode  major

```

1-1-1-2/gpononu inactive,bip threshold exceeded,dsa minor



Note: If more than one error condition is present (example: Excessive BIP errors and Optical Rx Power too high), the local alarm text may be too long to fit within the display output. In this case, AlarmType shows "issue 'onu status' command 0x300002" in order to prompt user to enter the **onu status** command for more details. The "0x300002" value is the actual alarm status word and will vary.

```
MXK-13> alarm show
*****      Central Alarm Manager      *****
ActiveAlarmCurrentCount      :8
AlarmTotalCount              :32
ClearAlarmTotalCount         :24
OverflowAlarmTableCount      :0
ResourceId      AlarmType      AlarmSeverity
-----
1-1-1-2/gpononu      issue 'onu status' command 0x300002      minor
```

GPON High and Low Receive Power Threshold Alarms

By default, the MXK will trigger a local alarm, and send a trap to ZMS when the GPON high/low receive power thresholds are crossed for the ONU received power on the upstream. The default value of the High threshold is -10 dbm. The default value of the Low threshold is -30 dbm. Users can change the default threshold values, and choose the upstream received power monitoring mode as desired.

ONU raises a “rx power out of range” alarm if us-rx-power-monitoring-mode in gpon-olt-onu-config profile is set to either monitorOnly or blockOnError and the alarm condition exists. When the alarm is set, the MXK will periodically restart the power level measurement. If the condition that cause the alarm is improved, a deactivated ONU is reactivated, and the alarm is cleared. The default interval for the periodic measurement is 5 minutes.

The GPON high/low receive power threshold values and monitoring modes are configured on a per-ONU basis with the **update gpon-olt-onu-config** command.

```
MXK-13> update gpon-olt-onu-config 1-1-1-2/gpononu
gpon-olt-onu-config 1-1-1-2/gpononu
Please provide the following: [quit].
serial-no-vendor-id: -----> {ZNTS}: ** read-only **
serial-no-vendor-specific: -----> {2216690777}: ** read-only **
password: -----> {}:
auto-learn: -----> {enabled}:
power-level: -----> {0}:
us-ber-interval: -----> {5000}:
ds-ber-interval: -----> {5000}:
onu-added: -----> {true}:
```

```

omci-file-name: -----> {}:
ONU-Managed-Entity-Profile-name: -----> {znid-gpon-2510-omci-4port-me}:
ONU-Generic-Assignments-Profile-name: -> {znid-gpon-2510-omci-4port-gen}:
physical-traps: -----> {disabled}:
ont-traps: -----> {disabled}:
line-status-traps: -----> {disabled}:
auto-upgrade: -----> {enabled}:
serial-no-vendor-specific-fsan: -----> {84200459}: ** read-only **
use-reg-id: -----> {disabled}:
us-rx-power-monitoring-mode: -----> {monitorOnly}:
us-rx-power-high-threshold: -----> {-10}:
us-rx-power-low-threshold: -----> {-30}:
dba-status-reporting: -----> {disabled}
.....
Save changes? [s]ave, [c]hange or [q]uit:q

```

Table 106: Received power threshold attributes in gpon-olt-onu-config profile

Attribute	Description
us-rx-power-monitoring-mode	Disable or enable the received power threshold alarm. Values: disabled This feature is disabled. monitorOnly Monitor ONU Receive Power Level. When ONU Receive Power Level crosses either the High or Low thresholds, trigger a local alarm, and send trap to ZMS. blockOnError Monitor ONU Receive Power Level. When ONU Receive Power Level crosses either the High or Low thresholds, trigger a local alarm, send trap to ZMS, disable ONT ranging and set ONT line status to DSA. Default: monitorOnly
us-rx-power-high-threshold	Upstream Receive Power High Threshold value, in the unit of dbm. Default: -10
us-rx-power-low-threshold	Upstream Receive Power Low Threshold value, in the unit of dbm. Default: -30

Configuring GPON high and low received power threshold alarms

- 1 View the ONU status. In this example, the upstream ONU received power under the OLT Rx Power column is -23.8 dBm, which is within the default value range of the GPON high and low received power threshold (-10 to -30).

```

MXK-13> onu status 1/1/2

```

ID	Onu	OperStatus	Omci ConfigState	Gpon OnuStatus	Download State	OLT Rx Power	ONT Rx Power	Distance (KM)
2	1-1-1-2	Up	Active	Active	NoUpgrade	-23.8 dBm	-23.0 dBm	18

- 2 Configure the upstream ONU received power monitoring mode and thresholds as desired.

This example changes the low-threshold to -20 from the default value -30, and changes the monitoring mode to **blockonerror**. If the current OLT RX power has crossed the low threshold, a received power threshold alarm will be triggered, and the ONU will be disabled.

```
MXK-13> update gpon-olt-onu-config 1-1-1-2/gpononu
gpon-olt-onu-config 1-1-1-2/gpononu
Please provide the following: [q]uit.
serial-no-vendor-id: -----> {ZNTS}: ** read-only **
serial-no-vendor-specific: -----> {2216690777}: ** read-only **
password: -----> {}:
auto-learn: -----> {enabled}:
power-level: -----> {0}:
us-ber-interval: -----> {5000}:
ds-ber-interval: -----> {5000}:
onu-added: -----> {true}:
omci-file-name: -----> {}:
ONU-Managed-Entity-Profile-name: -----> {znid-gpon-2510-omci-4port-me}:
ONU-Generic-Assignments-Profile-name: -> {znid-gpon-2510-omci-4port-gen}:
physical-traps: -----> {disabled}:
ont-traps: -----> {disabled}:
line-status-traps: -----> {disabled}:
auto-upgrade: -----> {enabled}:
serial-no-vendor-specific-fsan: -----> {84200459}: ** read-only **
use-reg-id: -----> {disabled}:
us-rx-power-monitoring-mode: -----> {monitorOnly}:blockonerror
us-rx-power-high-threshold: -----> {-10}:
us-rx-power-low-threshold: -----> {-30}:-20
dba-status-reporting: -----> {disabled}
.....
Save changes? [s]ave, [c]hange or [q]uit:s
```

- 3 View the ONU status.

This example shows GponOnuStatus is inactive and RXPWRDSA (i.e. received power out of range, and ONU is disabled.) Refer to the **alarm show** command for the explanation of the cryptic acronyms.

```
MXK-13> onu status 1/1/2
```

ONT ID	Distance (KM)	Omci OperStatus	Gpon ConfigState	Gpon OnuStatus	Download State	OLT Rx Power	OLT Rx Power
2	1-1-1-2 0.0	Down	Inactive	Inactive+RXPWRDSA	None	error	error

- 4 View the alarms on this ONU at the system level.

Two alarms are raised, link down and Rx power threshold alarms.

```
MXK-13> alarm show
```

```
*****      Central Alarm Manager      *****
ActiveAlarmCurrentCount      :8
AlarmTotalCount              :32
ClearAlarmTotalCount         :24
OverflowAlarmTableCount      :0

ResourceId      AlarmType      AlarmSeverity
-----
1-a-3-0/eth      linkDown      critical
1-1-1-0/gponolt  linkDown      critical
1-1-2-0/gponolt  linkDown      critical
1-1-3-0/gponolt  linkDown      critical
1-1-4-0/gponolt  linkDown      critical
1-1-1-2/gpononu  linkDown      minor
system          not_in_redundant_mode      major
1-1-1-2/gpononu  inactive,rx power out of range,dsa      minor
```

Rogue ONU detection and rogue ONU alarms

A rogue ONU is an ONU that transmits outside of its allocated bandwidth map. It can cause disruption to multiple subscribers or to all subscribers on a PON. Zhone provides versatile ways to detect a rogue ONU that is present on the PON and/or shut it down. That saves the other subscribers from experiencing any service issues.

To detect and/or shutdown a rogue ONU, use the following detection modes per OLT basis:

1. Periodical background process detection mode

Periodical background process detection mode can detect certain cases of rogue ONUs, but will not disable rogue ONUs. If a rogue ONU has been detected on the OLT, an OLT-level alarm “**gpon_olt_rogue_onu_detected**” is raised.

Refer to [Periodical background process detection mode on page 997](#) for the details.

2. Rogue RSSI detection mode

Rogue RSSI detection mode can detect and disable rogue ONUs by using the RSSI measurement on ONUs. If a rogue ONU has been detected on an OLT, an OLT-level alarm “**gpon_olt-rssi_rogue_onu_detected**” is raised. And then this rogue ONU will be identified, isolated, and disabled. An ONU-level alarm “**inactive, rogue ONU**” will be raised too.

Refer to [Rogue RSSI detection mode on page 1000](#) for the details.

3. Auto rogue RSSI detection mode

Auto rogue RSSI detection mode normally functions as disabled, it will be switched to the rogue RSSI detection mode only under certain circumstances.

Refer to [Auto rogue RSSI detection mode on page 1005](#) for the details.

All three kinds of rogue ONU alarms have severity levels as minor.



Note: As a rule users'd only want to use either **disabled** or **autorssi** mode under normal conditions, although users might want to set either **roguerssi** or **backgroundprocess** if users suspect a rogue ONU for some reasons that is not detected or not isolated by **autorssi** mode.

Users can configure the ONU detection modes in the gpon-olt-config profile. This profile contains three rogue ONU detection related attributes:

```
zSH> show gpon-olt-config
max-rt-propagation-delay:-----> {0 - 0}
max-onu-response-time:-----> {0 - 0}
preassigned-eqd:-----> {0 - 0}
los-alpha:-----> {0 - 0}
lof-alpha:-----> {0 - 0}
loam-alpha:-----> {0 - 0}
scrambler:-----> enabled disabled
fec-mode:-----> enabled disabled
auto-learn:-----> enabled disabled
power-level:-----> {0 - 0}
guard-bit-count:-----> {0 - 0}
dba-mode:-----> predictive piggyback
wholereport
gem-block-size:-----> {0 - 0}
us-ber-interval:-----> {0 - 0}
ds-ber-interval:-----> {0 - 0}
ber-sf-threshold:-----> {3 - 8}
ber-sd-threshold:-----> {4 - 9}
fec-request:-----> enabled disabled
key-exchange:-----> enabled disabled
min-rt-propagation-delay:-----> {0 - 0}
min-onu-response-time:-----> {0 - 0}
eqd-measure-cycles:-----> {0 - 0}
drift-ctrl-interval:-----> {0 - 0}
drift-ctrl-limit:-----> {0 - 0}
alloc-cycle-length:-----> {1 - 10}
min-us-alloc:-----> {0 - 0}
ack-timeout:-----> {0 - 0}
pls-max-alloc-size:-----> {0 - 0}
dba-cycle:-----> {2 - 10}
sr-dba-reporting-block-size:--> {0 - 0}
protection-switchover-timer:--> {0 - 0}
preamble-override:-----> enabled disabled
preamble-type-0:-----> {8}
preamble-type-1:-----> {8}
preamble-type-3-pre-range:----> {8}
preamble-type-3-post-range:---> {8}
preamble-type-3-pattern:-----> {8}
bip-error-monitoring-mode:----> disabled monitoronly
blockonerror
```



```

errors-per-sample-threshold:--> {0 - 0}
errored-samples-threshold:----> {0 - 0}
bip-max-sample-gap:-----> {0 - 0}
rogue-onu-detection:-----> disabled roguerssi
backgroundprocess autorssi
rogue-onu-detect-frequency:---> {1 - 60}
rogue-onu-rx-power-threshold:-> {0 - 0}

```

Table 107: rogue ONU detection attributes in gpon-olt-config profile

Attribute	Description
rogue-onu-detection	Disable or enable the rogue ONU detection modes. Values: disabled Disable all the rogue ONU detection mode. roguerssi Enable rogue RSSI detection. When a rogue ONU RSSI measurement crosses the rogue-onu-rx-power-threshold, an attempt is made to isolate the rogue ONU. If successful, disable the rogue ONU. Trigger a local alarm and send a trap to ZMS. backgroundprocess Enable background process detection. When a rogue transmission is detected, trigger a local alarm and send a trap to ZMS. autorssi Enable auto RSSI detection. In this mode, it normally stays as disabled, it will switch to the rogue RSSI detection mode 1) if more than half the active ONUs go down within a brief interval, 2) or if BIP errors exceed threshold on any ONU. Note that the second case is a detect only measurement, no attempt to disable the rogue ONU will be automatically performed. Default: disabled
rogue-onu-detect-frequency	How often to run a detection after enabling the detection. Default: 10 seconds
rogue-onu-rx-power-threshold	Upstream Receive Power High Threshold value for detecting rogue ONU, in the unit of dbm. RSSI upstream received power is measured on an unused ONU, which should measure zero, if the measurement exceeds the threshold, an alarm is reported and isolation is attempted. This is ignored in background process mode. Default: -30

Periodical background process detection mode

Certain rogue behaviors can only be detected by running the periodical background process detection mode on an OLT. This mode can only be used to detect the condition, rather than disable it.

The periodical background process opens a special allocation window and monitors for potential rogue transmission. The special window is opened with an unused Alloc_ID, for which no response is expected unless there is a rogue ONU. The window is opened so that it may detect a transmission either within the PON distance or further.

When a rogue ONU transmission is detected in the special window, “**gpon_olt_rogue_onu_detected**” alarm is raised on the OLT port. It shows

there is a rogue ONU has been detected on this OLT. The alarm severity level is minor.

Running the periodical background process detection and viewing OLT-level rogue ONU alarms

- 1 Set the rogue ONU detection mode to the periodical background process.

This example uses the default settings 10 seconds in the rogue-onu-detect-frequency field. After enabling the periodical background process, the process will run every 10 seconds.

```
zSH> update gpon-olt-config 1-1-1-0/gponolt
gpon-olt-config 1-1-1-0/gponolt
Please provide the following: [q]uit.
max-rt-propagation-delay: ----> {200}:
max-onu-response-time: -----> {50}:
preassigned-eqd: -----> {0}:
los-alpha: -----> {4}:
lof-alpha: -----> {4}:
loam-alpha: -----> {3}:
scrambler: -----> {enabled}:
fec-mode: -----> {disabled}:
auto-learn: -----> {enabled}:
power-level: -----> {0}:
guard-bit-count: -----> {32}:
dba-mode: -----> {predictive}:
gem-block-size: -----> {16}:
us-ber-interval: -----> {5000}:
ds-ber-interval: -----> {5000}:
ber-sf-threshold: -----> {3}:
ber-sd-threshold: -----> {5}:
fec-request: -----> {disabled}:
key-exchange: -----> {disabled}:
min-rt-propagation-delay: -----> {0}:
min-onu-response-time: -----> {10}:
eqd-measure-cycles: -----> {5}:
drift-ctrl-interval: -----> {1000}:
drift-ctrl-limit: -----> {3}:
alloc-cycle-length: -----> {2}:
min-us-alloc: -----> {16}:
ack-timeout: -----> {2000}:
pls-max-alloc-size: -----> {120}:
dba-cycle: -----> {2}:
sr-dba-reporting-block-size: --> {48}:
protection-switchover-timer: --> {500}:
preamble-override: -----> {disabled}:
preamble-type-0: -----> {0x00}:
preamble-type-1: -----> {0x00}:
preamble-type-3-pre-range: ----> {0x0b}:
preamble-type-3-post-range: ---> {0x08}:
preamble-type-3-pattern: -----> {0xaa}:
bip-error-monitoring-mode: ----> {monitoronly}:
errors-per-sample-threshold: --> {100}:
```

```

errored-samples-threshold: ----> {10}:
bip-max-sample-gap: -----> {10}:
rogue-onu-detection: ----->
{disabled}:backgroundprocess
rogue-onu-detect-frequency: ---> {10}:
rogue-onu-rx-power-threshold: -> {-30}:
.....

```

Save changes? [s]ave, [c]hange or [q]uit:**s**

- 2 If there are any rogue ONUs under this OLT port have been detected by running the periodical background process, the rogue ONU alarm will be raised on the OLT port.

Use the **alarm show** command to check the rogue ONU alarms:

```

zSH> alarm show
*****      Central Alarm Manager      *****
ActiveAlarmCurrentCount      :8
AlarmTotalCount              :15
ClearAlarmTotalCount         :7
OverflowAlarmTableCount      :0
ResourceId      AlarmType      AlarmSeverity
-----
1-a-3-0/eth      linkDown      critical
1-1-1-0/gponolt  gpon_olt_rogue_onu_detected  minor
...

```

Clearing OLT-level rogue ONU alarms

To clear the OLT level rogue ONU alarm “gpon_olt_rogue_onu_detected”, ONTs must be manually disabled one by one until the condition is no longer detected, at which time the alarm will go away.

If the rogue ONU detection mode is switched back to normal states, the alarm will clear, but condition still stay.

Switching the rogue ONU detection mode back to normal states from the periodical background process mode

After finished the periodical background process, follow the rule to set the rogue ONU detection mode back to the normal states. Normal states could be disabled or auto RSSI mode.

```

zSH> update gpon-olt-config 1-1-1-0/gponolt
gpon-olt-config 1-1-1-0/gponolt
Please provide the following: [q]uit.
max-rt-propagation-delay: ----> {200}:
max-onu-response-time: -----> {50}:
preassigned-eqd: -----> {0}:
los-alpha: -----> {4}:
lof-alpha: -----> {4}:
loam-alpha: -----> {3}:
scrambler: -----> {enabled}:

```

```

fec-mode: -----> {disabled}:
auto-learn: -----> {enabled}:
power-level: -----> {0}:
guard-bit-count: -----> {32}:
dba-mode: -----> {predictive}:
gem-block-size: -----> {16}:
us-ber-interval: -----> {5000}:
ds-ber-interval: -----> {5000}:
ber-sf-threshold: -----> {3}:
ber-sd-threshold: -----> {5}:
fec-request: -----> {disabled}:
key-exchange: -----> {disabled}:
min-rt-propagation-delay: -----> {0}:
min-onu-response-time: -----> {10}:
eqd-measure-cycles: -----> {5}:
drift-ctrl-interval: -----> {1000}:
drift-ctrl-limit: -----> {3}:
alloc-cycle-length: -----> {2}:
min-us-alloc: -----> {16}:
ack-timeout: -----> {2000}:
pls-max-alloc-size: -----> {120}:
dba-cycle: -----> {2}:
sr-dba-reporting-block-size: --> {48}:
protection-switchover-timer: --> {500}:
preamble-override: -----> {disabled}:
preamble-type-0: -----> {0x00}:
preamble-type-1: -----> {0x00}:
preamble-type-3-pre-range: ----> {0x0b}:
preamble-type-3-post-range: ---> {0x08}:
preamble-type-3-pattern: -----> {0xaa}:
bip-error-monitoring-mode: ----> {monitoronly}:
errors-per-sample-threshold: --> {100}:
errored-samples-threshold: ----> {10}:
bip-max-sample-gap: -----> {10}:
rogue-onu-detection: ----->
{backgroundprocess}:disabled or autorssi
rogue-onu-detect-frequency: ---> {10}:
rogue-onu-rx-power-threshold: -> {-30}:
.....

```

Save changes? [s]ave, [c]hange or [q]uit:**s**

Rogue RSSI detection mode



Caution: The rogue RSSI measurement is a semi-invasive mode. During the activation of the RSSI measurement on an OLT port it is not allowed to provision Alloc_IDs or to activate ONUs under that OLT port.

The rogue RSSI detection not only can detect the rogue ONU, and also can disable it. Note that after the rogue ONU had been disabled, this disabled ONU must be cleared and physically removed.

If the periodical background process detection cannot find the rogue ONU, users can run the rogue RSSI detection.

If users want to provision Alloc_IDs (by creating bridges/interfaces on ONU Gemports), and activate ONUs (by assigning serial numbers to ONUs ports), users must change the rogue-onu-detection mode from rogue RSSI to auto RSSI mode or disabled, after clearing any disabled ONUs.

A rogue RSSI detection performs the following two parts:

1. 1st part: Detect rogue ONUs and get OLT-level alarms

The rogue RSSI detection uses the rogue RSSI measurement to identify a rogue ONU by measuring transmission power on an unused ONU.

The RSSI measurement is a stand-alone utility for testing rogue transmission when no upstream burst is expected. The intention is to identify when a rogue ONU injects a constant energy on the link, and does not respond to OLT allocations.

If the rogue ONU RSSI measurement is higher than the rogue-onu-rx-power-threshold defined in the **gpon-olt-config** profile, an OLT-level alarm “**gpon_olt_rssi_rogue_onu_detected**” and trap will be sent. This rogue ONU alarm shows there is a rogue ONU has been detected with the RSSI measurement on the OLT.

2. 2nd part: Isolate and shutdown rogue ONUs, and get ONU-level alarms

Once the rogue RSSI detection is determined that a rogue ONU does exist on an OLT, it will start the process to determine which one is the rogue ONU and disable it. The process disables ONTs one at a time, each time it will perform a rogue RSSI measurement, until get a good reading, at which time it declare the last ONU disabled as rogue, and enable all the other “good” ONUs. An ONU error will be raised on the isolated ONU, and this ONU shows as disabled in the showline.

When this rogue ONU is disabled, an ONU-level alarm “**inactive, rogue onu**” and trap will be sent.

Note that the **2nd part** may fail, in which case the OLT-level alarm continues to be displayed. Reasons for failing to isolate rogue ONUs could be:

- Certain models of ONT do not disable transmission
- If a rogue ONU is connected with OLT on the fiber but not activated (by associating an ONU port ID with its serial number with the **onu set** command), it will not be isolated
- If the ONT is too “bad” to respond to a disable request.

Running the RSSI rogue ONT detection and viewing OLT-level and ONU-level RSSI rogue alarms

- 1** To enable the RSSI rogue ONT detection mode, change the rogue-onu-detection field of the gpon-olt-config profile to **roguerssi**.

This example uses the default values set in the rogue-onu-detect-frequency field and rogue-onu-rx-power-threshold field. That means the RSSI rogue ONT detection will run every 10 seconds, and if the RSSI measurement exceeds -30 dbm, an alarm is reported and isolation is attempted.

```

zSH> update gpon-olt-config 1-1-4-0/gponolt
gpon-olt-config 1-1-4-0/gponolt
Please provide the following: [q]uit.
max-rt-propagation-delay: -----> {200}:
max-onu-response-time: -----> {50}:
preassigned-eqd: -----> {0}:
los-alpha: -----> {4}:
lof-alpha: -----> {4}:
loam-alpha: -----> {3}:
scrambler: -----> {enabled}:
fec-mode: -----> {disabled}:
auto-learn: -----> {enabled}:
power-level: -----> {0}:
guard-bit-count: -----> {32}:
dba-mode: -----> {predictive}:
gem-block-size: -----> {16}:
us-ber-interval: -----> {5000}:
ds-ber-interval: -----> {5000}:
ber-sf-threshold: -----> {3}:
ber-sd-threshold: -----> {5}:
fec-request: -----> {disabled}:
key-exchange: -----> {disabled}:
min-rt-propagation-delay: -----> {0}:
min-onu-response-time: -----> {10}:
eqd-measure-cycles: -----> {5}:
drift-ctrl-interval: -----> {1000}:
drift-ctrl-limit: -----> {3}:
alloc-cycle-length: -----> {2}:
min-us-alloc: -----> {16}:
ack-timeout: -----> {2000}:
pls-max-alloc-size: -----> {120}:
dba-cycle: -----> {2}:
sr-dba-reporting-block-size: --> {48}:
protection-switchover-timer: --> {500}:
preamble-override: -----> {disabled}:
preamble-type-0: -----> {0x00}:
preamble-type-1: -----> {0x00}:
preamble-type-3-pre-range: ----> {0x0b}:
preamble-type-3-post-range: ---> {0x08}:
preamble-type-3-pattern: -----> {0xaa}:
bip-error-monitoring-mode: ----> {monitoronly}:
errors-per-sample-threshold: --> {100}:
errored-samples-threshold: ----> {10}:
bip-max-sample-gap: -----> {10}:
rogue-onu-detection: -----> {disabled}:roguerssi
rogue-onu-detect-frequency: ---> {10}:
rogue-onu-rx-power-threshold: -> {-30}:
.....

```

Save changes? [s]ave, [c]hange or [q]uit:s

- 2 If a rogue ONU is detected, users will see a rogue ONU alarm on the OLT port.

```
zSH> alarm show
***** Central Alarm Manager *****
ActiveAlarmCurrentCount      :8
AlarmTotalCount              :15
ClearAlarmTotalCount         :7
OverflowAlarmTableCount      :0
ResourceId      AlarmType      AlarmSeverity
-----
1-a-3-0/eth      linkDown      critical
1-1-4-0/gponolt  gpon_olt_rssi_rogue_onu_detected  minor
...
```

- 3 If a rogue ONU is isolated and disabled, users will see a rogue ONU alarm on the ONU port. This alarm will clear the OLT-level alarm listed in the previous step, unless there are more rogue ONUs under this OLT port, or failed to isolate and shutdown the detected rogue ONUs.

```
zSH> alarm show
***** Central Alarm Manager *****
ActiveAlarmCurrentCount      :8
AlarmTotalCount              :15
ClearAlarmTotalCount         :7
OverflowAlarmTableCount      :0
ResourceId      AlarmType      AlarmSeverity
-----
1-a-3-0/eth      linkDown      critical
1-1-4-1/gpononu  inactive,rogue onu  minor
...
```

Clearing OLT-level and ONU-level RSSI rogue alarms

To clear the ONU level RSSI rogue alarm “inactive, rogue onu”, the ONT must be physically removed from the network before using the **gpononu clear** command.

To clear the OLT-level RSSI rogue alarms, “gpon_olt_rssi_rogue_onu_detected”, use the same method shown in [Clearing OLT-level rogue ONU alarms, page 999](#).

Note that alarms can be cleared by changing detect mode or threshold, but this will not clear the condition.

Switching the rogue ONT detection back to normal states from the rogue RSSI mode

After users detected a rogue ONU on an OLT by running the rogue RSSI detection mode, the OLT port is in a restricted mode keeps users from

activating any ONUs connected to it. To go back to the normal states, perform the following steps:

- 1 Physically unplug the rogue ONU.
- 2 Clear the rogue ONU. This step will clear the onu-level “inactive, rogue onu” alarm too.

```
zSH> gpononu clear 1/4/1
```

- 3 Set the rogue ONU detection mode back to the normal states (disabled or auto RSSI mode).

```
zSH> update gpon-olt-config 1-1-4-0/gponolt
gpon-olt-config 1-1-4-0/gponolt
Please provide the following: [q]uit.
max-rt-propagation-delay: ----> {200}:
max-onu-response-time: -----> {50}:
preassigned-eqd: -----> {0}:
los-alpha: -----> {4}:
lof-alpha: -----> {4}:
loam-alpha: -----> {3}:
scrambler: -----> {enabled}:
fec-mode: -----> {disabled}:
auto-learn: -----> {enabled}:
power-level: -----> {0}:
guard-bit-count: -----> {32}:
dba-mode: -----> {predictive}:
gem-block-size: -----> {16}:
us-ber-interval: -----> {5000}:
ds-ber-interval: -----> {5000}:
ber-sf-threshold: -----> {3}:
ber-sd-threshold: -----> {5}:
fec-request: -----> {disabled}:
key-exchange: -----> {disabled}:
min-rt-propagation-delay: ----> {0}:
min-onu-response-time: -----> {10}:
eqd-measure-cycles: -----> {5}:
drift-ctrl-interval: -----> {1000}:
drift-ctrl-limit: -----> {3}:
alloc-cycle-length: -----> {2}:
min-us-alloc: -----> {16}:
ack-timeout: -----> {2000}:
pls-max-alloc-size: -----> {120}:
dba-cycle: -----> {2}:
sr-dba-reporting-block-size: --> {48}:
protection-switchover-timer: --> {500}:
preamble-override: -----> {disabled}:
preamble-type-0: -----> {0x00}:
preamble-type-1: -----> {0x00}:
preamble-type-3-pre-range: ----> {0x0b}:
preamble-type-3-post-range: ---> {0x08}:
preamble-type-3-pattern: -----> {0xaa}:
bip-error-monitoring-mode: ----> {monitoronly}:
errors-per-sample-threshold: --> {100}:
```



```

errored-samples-threshold: ----> {10}:
bip-max-sample-gap: -----> {10}:
rogue-onu-detection: -----> {roguerssi}:disabled
or autorssi
rogue-onu-detect-frequency: ---> {10}:
rogue-onu-rx-power-threshold: -> {-30}:
.....

Save changes? [s]ave, [c]hange or [q]uit:s

```

Auto rogue RSSI detection mode

The auto rogue ONU RSSI detection mode normally functions as disabled, it will be switched to the rogue RSSI detection mode only if one of the following two cases happened:

1. Case 1: If more than half the active ONUs on the OLT go down within a brief interval (currently 1 minute), then the disabled mode will be switched to the rogue RSSI detection mode for one measurement cycle.

In case 1, there are three possible outcomes:

No.1 The process did not detect presence of a rogue ONU. Then the detection mode will be switched back to the disabled mode, with a “no rogue ONU was detected” message shown on the console.

No.2 A rogue ONU was detected and isolated. In this case, the rogue RSSI mode is retained, so that the ONU stays disabled until it is cleared, or until the rogue detection mode is changed.

An ONU-level alarm “**inactive, rogue onu**” and trap will be sent.

No.3 Presence of rogue ONU is detected, but unable to isolate the ONU. In this case, the detection mode will be switched back to disabled.

An OLT-level alarm “**gpon_olt_rssi_rogue_onu_detected**” and trap will be sent.

2. Case 2: If BIP errors exceed threshold on any ONU, the disabled mode will be switched to rogue RSSI detection mode. But in this case, it is a “detect only” measurement. No attempt at isolation will be automatically performed, only **No.1** and **No.3** of the above outcomes are possible.

Running the Auto RSSI rogue ONT detection and viewing RSSI rogue ONU alarm on an ONU

- 1 To enable the auto RSSI rogue ONT detection mode, change the rogue-onu-detection field of the gpon-olt-config profile to **autorssi**.

```

zSH> update gpon-olt-config 1-1-4-0/gponolt
gpon-olt-config 1-1-4-0/gponolt
Please provide the following: [q]uit.
max-rt-propagation-delay: ----> {200}:
max-onu-response-time: -----> {50}:
preassigned-eqd: -----> {0}:
los-alpha: -----> {4}:

```

```
lof-alpha: -----> {4}:
loam-alpha: -----> {3}:
scrambler: -----> {enabled}:
fec-mode: -----> {disabled}:
auto-learn: -----> {enabled}:
power-level: -----> {0}:
guard-bit-count: -----> {32}:
dba-mode: -----> {predictive}:
gem-block-size: -----> {16}:
us-ber-interval: -----> {5000}:
ds-ber-interval: -----> {5000}:
ber-sf-threshold: -----> {3}:
ber-sd-threshold: -----> {5}:
fec-request: -----> {disabled}:
key-exchange: -----> {disabled}:
min-rt-propagation-delay: -----> {0}:
min-onu-response-time: -----> {10}:
eqd-measure-cycles: -----> {5}:
drift-ctrl-interval: -----> {1000}:
drift-ctrl-limit: -----> {3}:
alloc-cycle-length: -----> {2}:
min-us-alloc: -----> {16}:
ack-timeout: -----> {2000}:
pls-max-alloc-size: -----> {120}:
dba-cycle: -----> {2}:
sr-dba-reporting-block-size: --> {48}:
protection-switchover-timer: --> {500}:
preamble-override: -----> {disabled}:
preamble-type-0: -----> {0x00}:
preamble-type-1: -----> {0x00}:
preamble-type-3-pre-range: -----> {0x0b}:
preamble-type-3-post-range: ---> {0x08}:
preamble-type-3-pattern: -----> {0xaa}:
bip-error-monitoring-mode: -----> {monitoronly}:
errors-per-sample-threshold: --> {100}:
errored-samples-threshold: -----> {10}:
bip-max-sample-gap: -----> {10}:
rogue-onu-detection: -----> {disabled}:autorssi
rogue-onu-detect-frequency: ---> {10}:
rogue-onu-rx-power-threshold: -> {-30}:
.....
```

Save changes? [s]ave, [c]hange or [q]uit:**s**

2 If a rogue ONU is detected, users will see a rogue ONU alarm on the OLT port.

```
zSH> alarm show
***** Central Alarm Manager *****
ActiveAlarmCurrentCount      :8
AlarmTotalCount              :15
ClearAlarmTotalCount         :7
OverflowAlarmTableCount      :0
ResourceId      AlarmType      AlarmSeverity
```

-----	-----	-----
1-a-3-0/eth	linkDown	critical
1-1-4-0/gponolt	gpon_olt_rssi_rogue_onu_detected	minor
...		

- 3 If a rogue ONU is isolated and disabled, users will see a rogue ONU alarm on the ONU port. This alarm will clear the OLT-level alarm listed in the previous step, unless there are more rogue ONUs under this OLT port, or failed to isolate and shutdown the detected rogue ONUs.

```
zSH> alarm show
***** Central Alarm Manager *****
ActiveAlarmCurrentCount      :8
AlarmTotalCount              :15
ClearAlarmTotalCount         :7
OverflowAlarmTableCount      :0
ResourceId      AlarmType      AlarmSeverity
-----
1-a-3-0/eth      linkDown      critical
1-1-4-1/gpononu  inactive,rogue onu      minor
...
```

ONU Dying Gasp Alarms

When an ONU is power down, and the line-status-traps field in the gpon-olt-onu-config profile has been set to enabled or auto, then the lineStatusChange trap will be sent and the Dying Gasp Alarm will be raised.

Dying gasp alarm provides ONU information to the service provider when this ONU is about to power down. The dying gasp event is followed by an ONU down event.

If there is a link down alarm prior to receipt of dying gasp alarm, link down alarm will be cleared and dying gasp alarm will be created.

Viewing ONU dying gasp alarm on an ONU

- 1 To enable the ONU dying gasp alarm, change the line-status-traps field of the gpon-olt-onu-config profile to **auto** or **enabled**.

This example sets enabled for line-status-traps field:

```
zSH> update gpon-olt-onu-config line-status-traps=enabled 1-7-1-1/gpononu
gpon-olt-onu-config 1-7-1-1/gpononu
Record updated.
```

- 2 Create trap-destination profile to define a trap recipient the MXK will send traps to.

```
zSH> new trap-destination 1
trap-destination 1
Please provide the following: [q]uit.
trapdestination: -----> {0.0.0.0}: 172.16.80.39      the IP address of the SNMP trap server
communityname: -----> {}:
```

```

resendseqno: -----> {0}:
ackedseqno: -----> {0}:
traplevel: -----> {low}:
traptype: -----> {0}:
trapadminstatus: -----> {enabled}:
gatewaytrapserveraddr: -> {none}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```

3 Power down this ONU.

4 User will see Dying Gasp in the alarm or trap description:

a Send lineStatusChange trap with trap value as inactive | dyinggasp.

You can check the trap values in the trap recipient console in MIB browser.

b Raise alarm string as Dying Gasp received.

```

OCT 23 11:46:13: alert : 1/7/1025: alarm_mgr: _laMgrLogMsg(): l=295 :
tLineAlarm: 01: 7:01:01 Minor ONU Down
Line 1/7/1/1/gpononu CAUSE: Dying Gasp received

```

c Get the onuStatus as Inactive+DG:

```

zSH> onu status 7/1/1

```

Download ID	OLT Onu	ONT OperStatus	Distance ConfigState	Gpon State	AutoConfig Rx Power	Rx Power	(KM)	OnuStatus
1	1-7-1-1	Down	Inactive	None	error	error	error	Inactive+DG

Init

ONU Manual Reboot Alarms

When an ONU is rebooted manually from ZMS or MXK, and the line-status-traps in the gpon-olt-onu-config profile has been set to enabled or auto, then the lineStatusChange trap will be sent and the ONU manual reboot alarm will be raised.

Viewing ONU manual rebooted alarm on an ONU

1 To enable the ONU manual rebooted alarm, change the line-status-traps field of the gpon-olt-onu-config profile to **auto** or **enabled**.

This example sets enabled for line-status-traps field:

```

zSH> update gpon-olt-onu-config line-status-traps=enabled 1-7-1-1/gpononu
gpon-olt-onu-config 1-7-1-1/gpononu
Record updated.

```

2 Create trap-destination profile to define a trap recipient the MXK will send traps to.

```

zSH> new trap-destination 1

```

```

trap-destination 1
Please provide the following: [q]uit.
trapdestination: -----> {0.0.0.0}: 172.16.80.39           the IP address of the SNMP trap server
communityname: -----> {}:
resendseqno: -----> {0}:
ackedseqno: -----> {0}:
traplevel: -----> {low}:
traptype: -----> {0}:
trapadminstatus: -----> {enabled}:
gatewaytrapserveraddr: -> {none}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```

3 Reboot this ONU:

```
zSH> onu reboot 7/1/1
```

4 User will see ONU reboot in the alarm or trap description:

- a** Send lineStatusChange trap with trap value as inactive | onu rebooted.
You can check the trap values in the trap recipient console in MIB browser.
- b** Raise alarm string as onu rebooted.

```

OCT 23 09:52:32: alert   : 1/7/1025: alarm_mgr: _laMgrLogMsg(): l=295 :
tLineAlarm: 01: 7:01:01 Minor ONU Down
Line 1/7/1/1/gpononu CAUSE: ONU rebooted

```

c Get the onuStatus as Inactive+onuRebooted:

```

zSH> onu status 7/1/1

```

Download ID	OLT Onu	ONT OperStatus	Distance ConfigState	Gpon State	AutoConfig Rx Power	Rx Power	(KM)	OnuStatus
1	1-7-1-1	Down	Inactive	None	error	error	error	Inactive+onuRebooted Init

GPON Traps

- [View or change trap reporting status on an ONU, page 1010](#)
- [Change alarm severity for LineStatusTraps, page 1011](#)

View or change trap reporting status on an ONU

The conditions that cause asynchronous reporting traps can be controlled from the OLT through SNMP. The purpose of these controls is to reduce trap traffic between the MXK and ZMS to allow more information about critical or failing ONUs.

These three trap types are reported on an ONU:

- **phy (PhysicalTraps):** Includes the power status, battery status, and physical intrusion conditions as reported from the ONU through OMCI.

The options for the PhysicalTraps are:

- **enable:** The PhysicalTraps are sent.
- **disable:** The PhysicalTraps are not sent. Default value.

- **ont (OntTraps):** The status of LAN facing ports on the ONU (e.g. ethernet port LanLos).

The options for the OntTraps are:

- **enable:** The OntTraps are sent.
- **disable:** The OntTraps are not sent. Default value.

- **line (LineStatusTraps):** It is originated on the MXK, and reports the ONU line going up or down.

The options for the LineStatusTraps are:

- **enable:** The linkUp, linkDown, and lineStatusChange traps are sent.
- **disable:** The lineStatusTraps are not sent. Default value.
- **auto:** In this setting, the linkUp or linkDown traps are not sent, only the lineStatusChange trap is sent if the line is going down with dying gasp (presumably powered down), if there is a manual ONU reboot; or if the line is coming up.
- **linkonly:** Sends traps to set and clear ONU linkDown alarm only. Dying Gasp alarm is suppressed in this mode.

View the current reporting status of traps on ONU(s) with the **gpononu traps show** [slot[/olt[/onu]]command.

```
zSH> gpononu traps show 1/4/2
Slot 1 olt 4
ONU      Name      PhysicalTraps  OntTraps  LineStatusTraps
===  =====  =====  =====  =====
    2      1-1-4-2    enabled      disabled      auto
```

Change the current reporting status of traps on ONU 1/4/2 with the **gpononu traps enable|disable|auto|linkonly slot/olt/onu phy|ont|line** command.

Note that only LineStatusTraps (i.e. line) has auto and linkonly options.

```
zSH> gpononu traps disable 1/4/2 phy
zSH> gpononu traps linkonly 1/4/2 line
```

Verify the settings in the show command:

```
zSH> gpononu traps show 1/4/2
Slot 1 olt 4
ONU      Name      PhysicalTraps  OntTraps  LineStatusTraps
=== =====
2        1-1-4-2    disabled      disabled  linkonly
```

Change alarm severity for LineStatusTraps

Users can change the alarm severity for an ONT when the LineStatusTraps are sent. The LineStatusTraps includes linkUp, linkDown, and lineStatusChange traps.

To change the alarm severity of LineStatusTraps, use the "link-status-alarm-severity" field in the gpon-olt-onu-config profile. By default, the alarm severity is minor, it could be changed to major or critical.

The following example sets LineStatusTraps to auto, and sets the alarm severity level of LineStatusTraps to major:

```
zSH> update gpon-olt-onu-config 1-1-1-1
gpon-olt-onu-config 1-1-1-1/gpononu
Please provide the following: [q]uit.
serial-no-vendor-id: -----> {ZNTS}:  **
read-only **
serial-no-vendor-specific: -----> {51974624}:  **
read-only **
password: -----> {}:
auto-learn: -----> {enabled}:
power-level: -----> {0}:
us-ber-interval: -----> {5000}:
ds-ber-interval: -----> {5000}:
onu-added: -----> {true}:
omci-file-name: -----> {}:
ONU-Managed-Entity-Profile-name: -----> {zhone-2425}:
ONU-Generic-Assignments-Profile-name: -> {}:
physical-traps: -----> {disabled}:
ont-traps: -----> {disabled}:
line-status-traps: -----> {disabled}:
auto
auto-upgrade: -----> {enabled}:
serial-no-vendor-specific-fsan: -----> {031911E0}:  **
read-only **
use-reg-id: -----> {disabled}:
us-rx-power-monitoring-mode: -----> {monitoronly}:
us-rx-power-high-threshold: -----> {-10}:
```

```
us-rx-power-low-threshold: -----> {-30}:
dba-status-reporting: -----> {disabled}:
auto-config-state: -----> {init}: **
read-only **
link-status-alarm-severity: -----> {minor}: major
.....
Save changes? [s]ave, [c]hange or [q]uit: s
```


11

MXK VDSL2 CARDS

This chapter describes the MXK 24-port and 48-port VDSL2 cards and VDSL2 configuration:

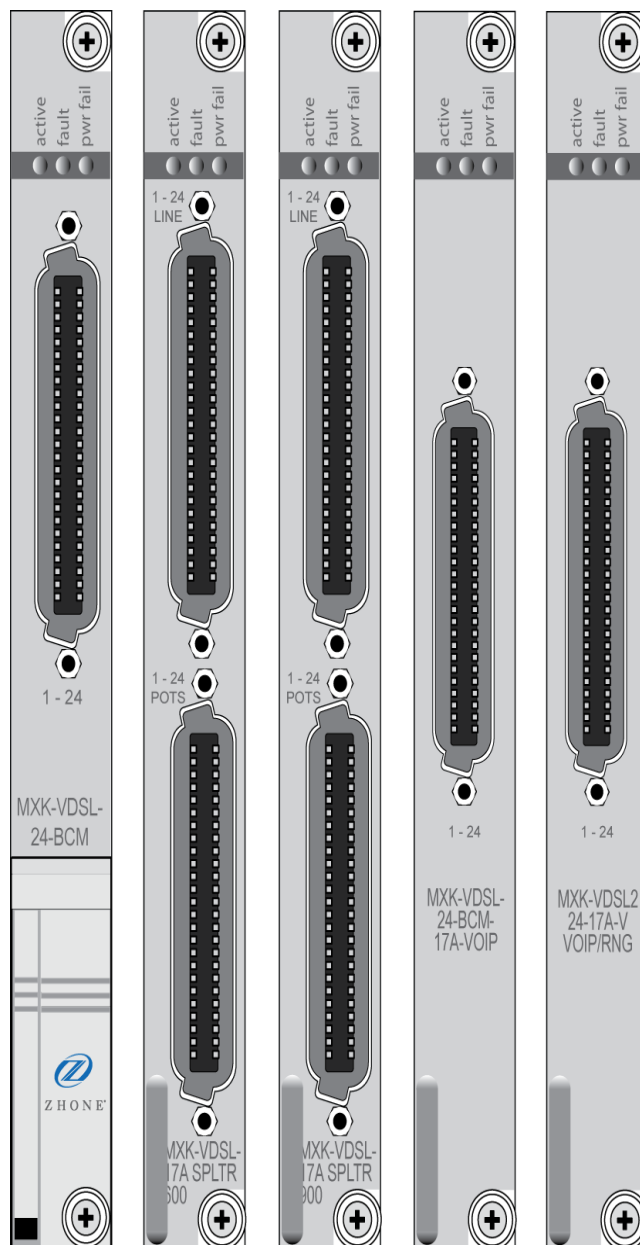
- [VDSL2 24-port single slot cards, page 1013](#)
- [VDSL2 48-port single slot card, page 1020](#)
- [VDSL2 on the MXK, page 1027](#)
- [VDSL2 interfaces, page 1029](#)
- [ADSL2+ fallback for VDSL2, page 1065](#)
- [ADSL2+ and VDSL2 bonding, page 1079](#)
- [Upstream Power Backoff \(UPBO\) for VDSL2, page 1104](#)
- [Downstream Power Backoff \(DPBO\), page 1106](#)
- [VDSL2 statistics, page 1116](#)
- [VDSL2 24-port card pinouts, page 1126](#)
- [VDSL2 48-port card pinouts, page 1127](#)

VDSL2 24-port single slot cards

This section describes the VDSL2 24-port line cards and how to configure VDSL2 interfaces.

- [VDSL2 24-port card overview, page 1014](#)
- [VDSL2 card specifications, page 1015](#)
- [VDSL2 24-port card configuration, page 1016](#)
- [View additional card information, page 1019](#)

VDSL2 24-port card overview



The VDSL2+ bond 24-port cards are:

- MXK-VDSL2-BCM-17A-24
- MXK-VDSL2--SPLTR600-BCM-17A-24
- MXK-VDSL2--SPLTR900-BCM-17A-24
- MXK-VDSL2-POTS-BCM-17A-24

MXK-VDSL2-BCM-17A-24

The MXK-VDSL2-24-BCM card is single-slot 24-port VDSL2 subscriber line card which provides high symmetric and asymmetric bandwidth and supports up to 17a profile.

MXK-VDSL2--SPLTR600-BCM-17A-24

MXK-VDSL2--SPLTR900-BCM-17A-24

These cards provide integrated POTS splitter to provide 24 ports of integrated VDSL2 and POTS service. Each of these lines are combined with the VDSL2 signal internally and exit the line card in the subscriber direction with both VDSL2 and POTS on the loop. In the network direction POTS is split from the VDSL2 signal keeping POTS on copper pairs and placing the VDSL2 data information on the IP network.

MXK-VDSL2-POTS-BCM-17A-24

This card provides 24 ports of integrated VDSL2 and POTS VoIP services. In addition to the standards listed in [Table 108](#), this card also supports SIP, SIP-PLAR, H.248, MGCP protocols and H.248 (MEGACO) protocols.

Configuration for the VDSL combo card providing POTS VoIP services is discussed in [Chapter 14, MXK POTS Cards](#).

Every VDSL2 card on the MXK can be used with Zhone VDSL and ADSL modems.

VDSL2 card specifications

Table 108: MXK-VDSL2-24 card specifications

Specification	Value
Size	Single slot
Density	24 ports Splitter cards have 24 VDSL2 ports plus 24 POTS ports
Physical interfaces	One (1) RJ-21X Champ 50-pin connector Two (2) RJ-21X Champ 50-pin connectors for splitter cards
Nominal line rates	Fast mode: - upstream: 58.177 Mbps - downstream: 100.160 Mbps Interleaved mode (minINP[2], maxDelay [20]): - upstream: 29.835 Mbps - downstream: 100.205 Mbps PhyR™ mode: - upstream: 18.232 Mbps - downstream: 73.44 Mbps

Table 108: MXK-VDSL2-24 card specifications (Continued)

Specification	Value
Power consumption	VDSL2-24 - Nominal: 32 W - Additional per DSL port: 1.0 W - Total Maximum: 56 W VDSL + splitter - Nominal: 32 W - Additional per DSL port: 1.0 W - Total Maximum: 56 W VDSL + POTS - Nominal: 38 W - Additional per DSL port: 1 W - Additional per POTS ports: 0.9 W - Total Maximum: 84 W
Standards supported	ITU G.993.2 (VDSL2 Annex A, B, L, and M) ANSI T1.413 Issue I and II ETSI TR328 and TS 101-270, ITU-T G.992.1 (Annex A, B, C, and I) G.992.2 (G.lite)) G.992.5 (Annex A, B, and M) G.993.2 (G.vdsl2) G.996.1(G.test) G997.1 (G.Ploam) IEEE 802.3ah 10-PASS-TS

VDSL2 24-port card configuration

Each card in the system must have a **card-profile**. The line card type determines the parameter settings in the **card-profile** and the software image for the card. Performing a **card add** <slot #> automatically creates the **card-profile** for the card with the correct software image and settings.

[Table 109](#) provides the type and the software image for the VDSL2 card on the MXK.

Table 109: VDSL2-24 card type and software image

Card	Type	Name of software image
MXK-VDSL2-24	10210	<i>mxlc24vdsl2.bin</i>
MXK-VDSL2-SPLTR600-BCM-17A-24	10210	<i>mxlc24vdsl2.bin</i>

Table 109: VDSL2-24 card type and software image

Card	Type	Name of software image
MXK-VDSL2-SPLTR900-BCM-17A-24	10210	<i>mxlc24vdsl2.bin</i>
MXK-VDSL2-POTS-BCM-17A-24	10222	<i>mxlc24vdsl2pots.bin</i>

Configuring a VDSL2 card

To configure a VDSL2 card on the MXK:

- 1 Install a VDSL2 card in the desired line card slot.
- 2 Create a **card-profile** for the card.

```
zSH> card add 1
```

After performing a **card add** in a slot, the slot resets and begins downloading the software image from the flash card. This could take a few moments.

When the card has finished loading, a log message similar to the following is displayed (if logging is enabled):

```
zSH> Card in slot slot-number changed state to RUNNING
```

- 3 Verify the card by entering **slots**:

```
zSH> slots
MXK 819
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
  1: MXK 24 PORT VDSL2 (RUNNING)
  5: MXK ADSL-48-B Bonded (RUNNING)
  6: MXK ADSL-48-B Bonded (RUNNING)
```

- 4 Verify the **card-profile** for the VDSL2 card:

```
zSH> get card-profile 1/1/10210
card-profile 1/1/10210
sw-file-name: -----> {mxlc24vdsl2.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {unknowntype}
card-atm-configuration: -> {notapplicable}
```

```

card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}

```

5 View card information including the state of the card and how long the card has been running:

```

zSH> slots 1
MXK 819
Type           : MXK 24 PORT VDSL2
Card Version   : 00001
EEPROM Version : 1
Serial #       : 2460301
CLEI Code      : No CLEI
Card-Profile ID : 1/1/10210
Shelf          : 1
Slot           : 1
ROM Version    : MXK 2.0.100
Software Version: MXK 2.0.117
State          : RUNNING
Mode           : FUNCTIONAL
Heartbeat check : enabled
Longest hbeat  : 102
Fault reset    : enabled
Uptime         : 6 hours, 7 minutes

```

Verifying the slot card installation

After saving the **card-profile** record, the slot card in that slot resets and downloads the software image from the flash card. This could take a few moments.

When the card finishes loading, a log message similar to the following is displayed (if logging is enabled):

```
zSH> Card in slot slot-number changed state to RUNNING
```

You can also use the **slots** command with the slot number of the card to view the state of the card and other information.

```

zSH> slots 1
MXK 819
Type           : MXK 24 PORT VDSL2
Card Version   : 00001
EEPROM Version : 1
Serial #       : 2460301
CLEI Code      : No CLEI
Card-Profile ID : 1/1/10210
Shelf          : 1
Slot           : 1
ROM Version    : MXK 2.0.100
Software Version: MXK 2.0.117
State          : RUNNING

```

```

Mode           : FUNCTIONAL
Heartbeat check : enabled
Longest hbeat  : 102
Fault reset    : enabled
Uptime         : 6 hours, 7 minutes

```

To view the status of all the cards, use the **slots** command without any arguments:

```

zSH> slots
MXK 819
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
  1: MXK 24 PORT VDSL2 (RUNNING)
  5: MXK ADSL-48-B Bonded (RUNNING)
  6: MXK ADSL-48-B Bonded (RUNNING)

```

View additional card information

View the EPROM version of the card:

```

zSH> eeshow card 1
EEPROM contents: for slot 1
EEPROM_ID      : 00 -- CARD
Version        : 01
Size           : 054
CardType       : 10210 -- MXLC24VDSL2
CardVersion    : 00001
SerialNum      : 02460301
ShelfNumber    : 00001
CLEI Code      : No CLEI
Cksum          : 0x2922

```

View the EPROM version of the daughter card:

```

zSH> eeshow 1d 1
EEPROM contents: for slot 1
EEPROM_ID      : 01 -- 1DAUGHTER
Version        : 01
Size           : 022
CardType       : 10210 -- MXLC24VDSL2
CardVersion    : 00001
SerialNum      : 02460317
ShelfNumber    : 00001
CLEI Code      : No CLEI
Cksum          : 0x2812

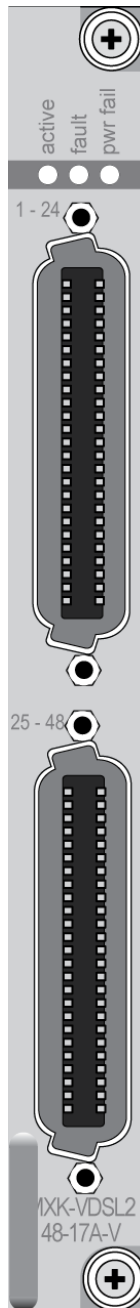
```

VDSL2 48-port single slot card

This section describes the VDSL2 48-port line card and includes:

- [VDSL2 48-port line card overview, page 1021](#)
- [VDSL2 48-port with vectoring, page 1021](#)
- [VDSL2 48-port card specifications, page 1022](#)
- [VDSL2 48-port card configuration, page 1023](#)
- [Cabling for the VDSL2 48-port card, page 1025](#)

VDSL2 48-port line card overview



MXK-VDSL2-BCM-17A-48-V

The MXK-VDSL2-BCM-17A-48-V card is single-slot 48-port VDSL2 subscriber line card which provides high symmetric and asymmetric bandwidth and supports 8a, 8b, 8c, 8d, 12a, 12b, and 17a profiles, ADSL2 fallback, link bonding and vectoring along with support for Phy-R for noise protection retransmission.

This VDSL2 48-port card supports board-level vectoring (BLV), which means that the vector group consists of VDSL2 ports on a single line card.

VDSL2 48-port with vectoring

This VDSL2 48-port card with vectoring provides a noise-canceling technology that reduces the noise on VDSL2 lines in a bundle allowing the line to operate at peak speeds. Noise-cancelling DSP technology is used to eliminate near-end and far-end crosstalk.

VDSL2 48-port card specifications

Table 110: MXK-VDSL2-48 card specifications

Specification	Value
Size	Single slot
Density	48 ports
Physical interfaces	Two (2) RJ-21X Champ 50-pin connector
Nominal line rates	Fast mode: - upstream: 58.177 Mbps - downstream: 100.160 Mbps Interleaved mode (minINP[2], maxDelay [20]): - upstream: 29.835 Mbps - downstream: 100.205 Mbps PhyR™ mode: - upstream: 18.232 Mbps - downstream: 73.44 Mbps
Power consumption	VDSL2-48 Nominal: 29 W Additional per DSL port: 0.51 W Total Maximum: 53 W
Standards supported	ANSI T1.413 Issue I and II ETSI TR 328 and TS 101-270 ITU-T G.992.1 (Annex A, B, C, and I) ITU-T G.992.2 (G.lite) ITU-T G.992.3 (Annex A, B, J, L, and M) ITU-T G.992.5 (Annex A, B, J, and M) ITU-T G.993.2 (G.VDSL2) ITU-T G.993.5 (G.vector) ITU-T G.996.1 (G.test) ITU-T G.997.1 (G.PLOAM) ITU-T G.998.4 (G.inp) ITU-T G.999.1 (T.int) G.bond/T1E1.4/2004-334R4 Provides support for IEEE 802.3ah 10-PASS-TS

VDSL2 48-port card configuration

Each card in the system must have a **card-profile**. The line card type determines the parameter settings in the **card-profile** and the software image for the card. Performing a **card add** <slot #> automatically creates the **card-profile** for the card with the correct software image and settings.

Table 109 provides the type and the software image for the VDSL2 48-port card on the MXK.

Table 111: VDSL2-48 port card type and software image

Card	Type	Name of software image
MXK-VDSL2-BCM-17A-48-V	10224	<i>mxlc48vdsl2.bin</i>

Configuring a VDSL2 48-port card

To configure a VDSL2 48-port card on the MXK:

- 1 Install a VDSL2 48-port card in the desired line card slot.
- 2 Create a **card-profile** for the card.

```
zSH> card add 1
```

After performing a **card add** in a slot, the slot resets and begins downloading the software image from the flash card. This could take a few moments.

When the card has finished loading, a log message similar to the following is displayed (if logging is enabled):

```
zSH> Card in slot slot-number changed state to RUNNING
```

- 3 Verify the card by entering **slots**:

```
zSH> slots
MXK 819
Uplinks
  a: *MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
  1: MXK 48 PORT VDSL2 (RUNNING)
  5: MXK ADSL-48-B Bonded (RUNNING)
  6: MXK ADSL-48-B Bonded (RUNNING)
```

- 4 Verify the **card-profile** for the VDSL2 48-port card:

```
zSH> get card-profile 1/1/10224
card-profile 1/1/10224
sw-file-name: -----> {mxlc48vdsl2.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
```

```

sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {unknowntype}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}

```

5 View card information including the state of the card and how long the card has been running:

```

zSH> slots 1
MXK 819
Type           : MXK 48 PORT VDSL2
Card Version    : 00001
EEPROM Version  : 1
Serial #        : 2460301
CLEI Code       : No CLEI
Card-Profile ID : 1/1/10224
Shelf           : 1
Slot            : 1
ROM Version     : MXK 2.0.100
Software Version: MXK 2.4.117
State           : RUNNING
Mode            : FUNCTIONAL
Heartbeat check : enabled
Longest hbeat   : 102
Fault reset     : enabled
Uptime          : 6 hours, 7 minutes

```

Verifying the slot card installation

After saving the **card-profile** record, the slot card in that slot resets and downloads the software image from the flash card. This could take a few moments.

When the card finishes loading, a log message similar to the following is displayed (if logging is enabled):

```
zSH> Card in slot slot-number changed state to RUNNING
```

You can also use the **slots** command with the slot number of the card to view the state of the card and other information.

```

zSH> slots 1
MXK 819
Type           : MXK 48 PORT VDSL2
Card Version    : 00001
EEPROM Version  : 1

```

```

Serial #           : 2460301
CLEI Code          : No CLEI
Card-Profile ID    : 1/1/10210
Shelf              : 1
Slot               : 1
ROM Version        : MXK 2.0.100
Software Version   : MXK 2.4.117
State              : RUNNING
Mode               : FUNCTIONAL
Heartbeat check    : enabled
Longest hbeat      : 102
Fault reset        : enabled
Uptime             : 6 hours, 7 minutes

```

To view the status of all the cards, use the **slots** command without any arguments:

```

zSH> slots
MXK 819
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
  1: MXK 48 PORT VDSL2 (RUNNING)
  5: MXK ADSL-48-B Bonded (RUNNING)
  6: MXK ADSL-48-B Bonded (RUNNING)

```

Cabling for the VDSL2 48-port card

The VDSL2 48-port card uses Amphenol connectors at a 120 degree angle as shown in [Figure 127](#). See [VDSL2 48-port card pinouts on page 1127](#) for pinout information and [Securing amphenol connectors on page 72](#) of the *MXK Hardware Installation Guide* for how to secure the Amphenol connectors.

Figure 127: VDSL2 48-port Amphenol 120 degree connectors



Note: Zhone recommends using a CAT 5 cable for at least the first 30 feet.

VDSL2 on the MXK

VDSL2 overview

VDSL2 functionality over copper wires is similar to ADSL2+, with some key distinctions. Currently, ADSL2+ is the most widely deployed access technology to provide high speed data from the central office. ADSL2+ utilizes bandwidth up to 2.2MHz, with operating speeds of approximately 28Mbps downstream and 1.1Mbps upstream (2.2Mbps upstream with Annex A implemented).

In contrast, VDSL2 utilizes up to 30 MHz of bandwidth to provide speeds of 100 Mbps, downstream and upstream, within 1000 ft. Data rates in excess of 25 Mbps are available for distances up to 4000 ft.

In spite of the distance constraints imposed by VDSL2, there is a notable exception. Additional opportunities for VDSL2 exist as telephone companies replace much of their main feeds with fiber such as fiber to the curb or fiber to the neighborhood. Placing a VDSL2 transceiver in the home and a VDSL2 MXK in the equipment box to leverage the fiber overcomes the distance constraints that restricts VDSL2 past 1000 ft.

VDSL2 standards

Standardized as ITU G.993.2, VDSL2 is an enhancement to G.993.1 (VDSL). Because the first generation of VDSL supported DMT (Discrete Multi-Tone modulation) in the main body and QAM (Quadrature amplitude modulation) in a normative annex, VDSL2 was only specified to support DMT modulation. This means that since the underlying DMT modulation code is the same as ADSL and ADSL2+, VDSL2 is fully compatible with existing services and enables backward-interoperability with ADSL.

DMT divides the available bandwidth into individual tones, each 4.3125KHz wide. Each tone can be thought of as a single transmitter capable of transmitting up to 15 bits of information each. DMT is able to monitor the SNR of each tone thus providing the ability to move data from tones that have a low SNR to tones that have a higher SNR and are able to handle the data error free.

Leveraging existing copper infrastructure utilized to provide POTS and ADSL/ADSL2+ service, VDSL2 can deliver up to 15% improvement over ADSL2+ by avoiding ATM cell overhead. Using ADSL2+ based services requires provisioning ATM based bridges for those ports since ADSL2+ standard uses ATM as its underlying protocol while VDSL2 employs PTM.

VDSL2 transmission

[Table 112](#) describes the profiles currently available for VDSL2.

Table 112: Profiles by region

Profiles	8a	8b	8c	8d	12a	12b	17a	30a
Bandwidth MHz	8.5				12		17.7	30
Bandwidth KHz	4.312	4.312	4.312	4.312	4.312	4.312	4.312	8.625
Tones D/S	1971				2770		4095	2098
TX Power D/S dBm	+17.5	+20.5	+11.5	+14.5				
Maximum throughput (Mbps)	50				68		100	200

VDSL2 on the MXK

The MXK's VDSL2 implementation provides full-rate VDSL2 performance for packet-based service delivery. This is a low-power, fully programmable, high performance solution that gives service providers a cost-effective way to provide triple-play service, including the different configurations of emerging video services. As an end-to-end solution, the MXK VDSL2 card works with VDSL2 CPEs to provide high-speed, long reach, and low cost solutions for multi-tenant and multi-dwelling units (MxU), as well as remote terminal and CO deployments.



Note: The high frequencies used in VDSL make the technology more susceptible to cross talk than ADSL. Because of this, Zhone recommends using CAT-5 rated cables for at least the first 30 feet from the line card. The tighter twists of CAT-5 cable greatly reduces the chance of cross-talk interference with VSDL.

The MXK also features ADSL2+ fallback mode implementation that supports one single VPI/VCI (0, 35) and is not configurable. The ADSL2+ CPE must be configured to use 0, 35.

VDSL2 interfaces

This section covers:

- [VDSL2 interface profiles, page 1029](#)
- [vdsl-config default parameters, page 1029](#)
- [vdsl-co-config default parameters, page 1034](#)
- [View vdsl-cpe-config profile default parameters, page 1041](#)
- [Configure VDSL2 profiles to cap train rates, page 1048](#)
- [Configure VDSL2 G.INP, page 1049](#)
- [VDSL2 statistics, page 1116](#)

VDSL2 interface profiles

VDSL2 interface configuration consists of three profiles:

- **vdsl-config**
- **vdsl-co-config**
- **vdsl-cpe-config**

vdsl-config default parameters

The parameter defaults set in the **vdsl-config** profile are appropriate for most configurations. When necessary, the **vdsl-config** parameters can be updated.

Viewing vdsl-config profile defaults

- 1 Verify the VDSL2 cards.

```
zSH> slots
MXK 819
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
Cards
  2: MXK 24 PORT VDSL2 (RUNNING)
  4: MXK 4 PORT GPON (RUNNING)
  5: MXK 8 PORT GPON (RUNNING)
  6: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
  10: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (RUNNING)
```

- 2 View the **vdsl-config** parameters and values.

```
zSH> show vdsl-config
transmit-mode:-----> autonegotiatemode vdsl2mode adsl2plusmode
adsl2mode gdmtdmode
line-type:-----> fastonly interleavedonly
```

```

vdsl2-profile:-----> g993-2-8a g993-2-8b g993-2-8c g993-2-8d
g993-2-12a g993-2-12b g993-2-17a g993-2-30a
adslAnnexMModeEnabled:-----> true false
adslAnnexMPsdMask:-----> eu64 eu60 eu56 eu52 eu48 eu44 eu40 eu36
eu32
trellis-enabled:-----> true false
rs-enabled:-----> true false
psd-shape:-----> region-a-nus0 region-a-eu-32 region-a-eu-36
region-a-eu-40 region-a-eu-44 region-a-eu-48 region-a-eu-52 region-a-eu-56
region-a-eu-60 region-a-eu-64 region-a-eu-128 region-a-adlu-32
region-a-adlu-36 region-a-adlu-40 region-a-adlu-44 region-a-adlu-48
region-a-adlu-52 region-a-adlu-56 region-a-adlu-60 region-a-adlu-54
region-a-adlu-128 region-b-998-m1x-a region-b-998-m1x-b region-b-998-m1x-nus0
region-b-998-m2x-a region-b-998-m2x-m region-b-998-m2x-b region-b-998-m2x-nus0
region-b-998-e17-m2x-nus0 region-b-998-e17-m2x-nus0-m
region-b-998-ade17-m2x-nus0-m region-b-998-ade17-m2x-a region-b-998-ade17-m2x-b
region-b-998-e30-m2x-nus0 region-b-998-e30-m2x-nus0-m
region-b-998-ade30-m2x-nus0-m region-b-998-ade30-m2x-nus0-a region-b-997-m1c-a-7
region-b-997-m1x-m-8 region-b-997-m1x-m region-b-997-m2x-m-8 region-b-997-m2x-a
region-b-997-m2x-m region-b-997-hpe17-m1-nus0 region-b-997-hpe30-m1-nus0
region-b-997-e17-m2x-a region-b-997-e30-m2x-nus0 region-b-997-bt-anfp
region-c-138-b region-c-276-b region-c-138-co region-c-276-co region-c-tcmisdn
region-c-1104-co-17a region-c-1104-co-30a
fallbackDefaultVpi:-----> {0 - 255}
fallbackDefaultVci:-----> {0 - 65535}

```

3 View the current **vdsl-config** default parameters for a VDSL2 port.

```

zSH> get vdsl-config 1-2-1-0/vdsl1
vdsl-config 1-2-1-0/vdsl1
transmit-mode: -----> {autonegotiatemode}
line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-17a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {35}

```

[Table 113](#) defines **vdsl-config** profile parameters.

Table 113: vdsl-config parameters

Parameter	Definition
transmit-mode	The VDSL2 transmission standard to be used for the line. Supported values are: autoNegotiateMode: automatically negotiates all supported transmission modes. vdslMode: The VDSL standards supported are: standard long-reach-8k r8k std-8k lr-8k std-lr std-lr-8k vdsl2Mode: The VDSL2 standards supported are: g993-2-8a(1), g993-2-8b g993-2-8c g993-2-8d g993-2-12a g993-2-12b g993-2-17a g993-2-30a adsl2Mode: The modem negotiates rates up to G.992.3 and G.992.4 (ADSL2). adsl2PlusMode: The modem negotiates rates up to G.992.5 (ADSL2+). gdmtdMode: G.dmt Default: autoNegotiateMode
line-type	Specifies the channelization of the VDSL2 line. fastonly No impulse noise protection, but lowest possible latency. Recommended only where lowest possible latency is required (for example, gaming). interleaveonly Better impulse noise protection with higher latency. Recommended for all voice, video, and/or data deployments. Default: interleaveonly

Table 113: vdsl-config parameters (Continued)

Parameter	Definition
vdsl2-profile	The VDSL2 standard to be used for the line. Standards: g993-2-8a g993-2-8b g993-2-8c g993-2-8d g993-2-12a g993-2-12b g993-2-17a g993-2-30a Default: g993-2-17a
adslAnnexMModeEnabled	G.992.3/4 with Annex M true false Default: false
adslAnnexMPsdMask	Specifies the maximum transmit PSD allowed in the downstream channel on an BIS Annex-M line. eu64 eu60 eu56 eu52 eu48 eu44 eu40 eu36 eu32 Default: eu32
trellis-enabled	Enables/disables Trellis coding. true false Default: true
rs-enabled	Enables or disables RS coding. true false Default: true

Table 113: vdsl-config parameters (Continued)

Parameter	Definition
psd-shape	Maximum transmit PSD allowed on a VDSL2 line. region-a-nus0, region-a-eu-32, region-a-eu-36, region-a-eu-40, region-a-eu-44, region-a-eu-48, region-a-eu-52, region-a-eu-56, region-a-eu-60, region-a-eu-64, region-a-eu-128, region-a-adlu-32, region-a-adlu-36, region-a-adlu-40, region-a-adlu-44, region-a-adlu-48, region-a-adlu-52, region-a-adlu-56, region-a-adlu-60, region-a-adlu-64, region-a-adlu-128, region-b-998-m1x-a, region-b-998-m1x-b, region-b-998-m1x-nus0, region-b-998-m2x-a, region-b-998-m2x-m, region-b-998-m2x-b, region-b-998-m2x-nus0, region-b-998-e17-m2x-nus0, region-b-998-e17-m2x-nus0-m, region-b-998-ade17-m2x-nus0-m, region-b-998-ade17-m2x-a, region-b-998-ade17-m2x-b, region-b-998-e30-m2x-nus0, region-b-998-e30-m2x-nus0-m, region-b-998-ade30-m2x-nus0-m, region-b-998-ade30-m2x-nus0-a, region-b-997-m1c-a-7, region-b-997-m1x-m-8, region-b-997-m1x-m, region-b-997-m2x-m-8, region-b-997-m2x-a, region-b-997-m2x-m, region-b-997-hpe17-m1-nus0, region-b-997-hpe30-m1-nus0, region-b-997-e17-m2x-a, region-b-997-e30-m2x-nus0 Default: region-a-eu-32
fallbackDefaultVpi	ATM VPI to be used by the line when it trains in ADSL/ATM mode and bridges are configured for single-vc mode. Default: 0
fallbackDefaultVci	ATM VCI to be used by the line when it trains in ADSL/ATM mode and bridges are configured for single-vc mode. Default: 35

Modifying the default vdsl-config parameters

Phy-R™ requires that the **line-type** parameter is set to *interleavedonly*.

- 1 Enter the **update vdsl-config interface/type** command.

```

zSH> update vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}:
line-type: -----> {fastonly}: interleavedonly
vdsl2-profile: -----> {g993-2-17a}:
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
fallbackDefaultVpi: -----> {0}:
fallbackDefaultVci: -----> {35}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

2 View the changed parameter.

```
zSH> get vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
transmit-mode: -----> {autonegotiatemode}
line-type: -----> {interleavedonly}
vdsl2-profile: -----> {g993-2-17a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {35}
```

vdsl-co-config default parameters

The VDSL2 downstream interface is the **vdsl-co-config** profile which defines downstream behavior.

The Broadcom Phy-R™ parameter **support** in the **vdsl-co-config** profile is enabled by default. The **line-type** parameter in the **vdsl-config** profile must be set to *interleavedonly* for Phy-R™ to work. See [Modifying the default vdsl-config parameters on page 1033](#).

Viewing vdsl-co-config profile defaults

View the **vdsl-co-config** parameters and their default values.

1 Verify the VDSL2 cards.

```
zSH> slots
MXK 819
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
  1: MXK 24 PORT VDSL2 (RUNNING)
  5: MXK ADSL-48-B Bonded (RUNNING)
  6: MXK ADSL-48-B Bonded (RUNNING)
```

2 View the vdsl-co-config parameters and values.

```
zSH> show vdsl-co-config
fastMaxTxRate:-----> {0 - 100000}
fastMinTxRate:-----> {0 - 100000}
interleaveMaxTxRate:-----> {0 - 100000}
interleaveMinTxRate:-----> {0 - 100000}
rateMode:-----> manual adapt-at-init dynamic
maxPower:-----> {-50 - 200}
maxSnrMgn:-----> {0 - 310}
minSnrMgn:-----> {0 - 310}
targetSnrMgn:-----> {0 - 310}
downshiftSnrMgn:-----> {0 - 310}
```

```

upshiftSnrMgn:-----> {0 - 310}
minDownshiftTime:-----> {0 - 16383}
minUpshiftTime:-----> {0 - 16383}
bitSwap:-----> disabled enabled
minINP:-----> noprotection halvesymbol singlesymbol
twosymbols threesymbols foursymbols fivesymbols sixsymbols sevensymbols
eightsymbols ninesymbols tensymbols elevensymbols twelvesymbols
thirteensymbols fourteensymbols fifteensymbols sixteensymbols
maxInterleaveDelay:-----> {0 - 63}
phyRSupport:-----> enable disable
phyRmaxINP:-----> {0 - 160}
phyRminRSoverhead:-----> {0 - 255}
phyRRtxRatio:-----> {0 - 255}
ginpVdslCoSupport:-----> enable disable
ginpVdslCoEtrMax:-----> {64 - 200000}
ginpVdslCoEtrMin:-----> {64 - 200000}
ginpVdslCoNdrMax:-----> {64 - 200000}
ginpVdslCoShineRatio:-----> {0 - 255}
ginpVdslCoLeftrThreshold:-----> {0 - 99}
ginpVdslCoMaxDelay:-----> {0 - 63}
ginpVdslCoMinDelay:-----> {0 - 63}
ginpVdslCoMin:-----> {0 - 127}
ginpVdslCoMinRSoverhead:-----> {0 - 64}
ginpVdslCoReinCfg:-----> {0 - 7}
ginpVdslCoReinFreq:-----> freq100hz freq120hz
ginpVdslCoRtxMode:-----> forbidden preferred forced testmode

```

3 View the current **vdsl-co-config** default parameters for a VDSL2 port.

```

zSH> get vdsl-co-config 1-2-1-0/vdsl
vdsl-co-config 1-2-1-0/vdsl
fastMaxTxRate: -----> {100000}
fastMinTxRate: -----> {0}
interleaveMaxTxRate: -----> {100000}
interleaveMinTxRate: -----> {0}
rateMode: -----> {dynamic}
maxPower: -----> {145}
maxSnrMgn: -----> {160}
minSnrMgn: -----> {0}
targetSnrMgn: -----> {60}
downshiftSnrMgn: -----> {30}
upshiftSnrMgn: -----> {90}
minDownshiftTime: -----> {30}
minUpshiftTime: -----> {30}
bitSwap: -----> {enabled}
minINP: -----> {twosymbols}
maxInterleaveDelay: -----> {20}
phyRSupport: -----> {enable}
phyRmaxINP: -----> {0}
phyRminRSoverhead: -----> {0}
phyRRtxRatio: -----> {0}
ginpVdslCoSupport: -----> {disable}
ginpVdslCoEtrMax: -----> {100000}
ginpVdslCoEtrMin: -----> {64}

```

```

ginpVdslCoNdrMax: -----> {100000}
ginpVdslCoShineRatio: -----> {10}
ginpVdslCoLeftrThreshold: -----> {0}
ginpVdslCoMaxDelay: -----> {20}
ginpVdslCoMinDelay: -----> {0}
ginpVdslCoMin: -----> {4}
ginpVdslCoMinRSoverhead: -----> {0}
ginpVdslCoReinCfg: -----> {0}
ginpVdslCoReinFreq: -----> {freq120hz}
ginpVdslCoRtxMode: -----> {preferred}

```

Table 114 defines the parameter values for the **vdsl-co-config** profile.

Table 114: vdsl-co-config parameter definitions

Parameter	Description
fastMaxTxRate	Specifies the maximum downstream fast channel data rate in steps of 1000 bits/second. Default: 200,000
fastMinTxRate	Specifies the minimum downstream fast channel data rate in steps of 1000 bits/second. Default: 0
interleaveMaxTxRate	Specifies the maximum downstream slow channel data rate in steps of 1000 bits/second. The maximum aggregate downstream transmit speed of the line can be derived from the sum of maximum downstream fast and slow channel data rates. Default: 200,000
interleaveMinTxRate	Specifies the minimum downstream slow channel data rate in steps of 1000 bits/second. The minimum aggregate downstream transmit speed of the line can be derived from the sum of minimum downstream fast and slow channel data rates. Default: 0
rateMode	Specifies the rate selection behavior for the line in the downstream direction. manual: forces the rate to the configured rate adapt-at-init: adapts the line at initialization only dynamic: adapts the line at initialization and showtime Default: dynamic
maxPower	Specifies the maximum aggregate downstream power level in the range 0 to 14.5 dBm. Default: 200
maxSnrMgn	Specifies the maximum downstream Signal/Noise Margin in units of 0.10 dB, for a range of 0 to 31.0 dB. Default: 127

Table 114: vdsi-co-config parameter definitions (Continued)

Parameter	Description
minSnrMgn	Specifies the minimum downstream Signal/Noise Margin in units of 0.10 dB, for a range of 0 to 31.0 dB. Default: 0
targetSnrMgn	Specifies the target downstream Signal/Noise Margin in units of 0.10 dB, for a range of 0 to 31.0 dB. This is the Noise Margin the transceivers must achieve with a BER of 10^{-7} or better to successfully complete initialization. Default: 60
downshiftSnrMgn	Configured Signal/Noise Margin for rate downshift. If the noise margin falls below this level, the modem should attempt to decrease its transmit rate. Default: 30
upshiftSnrMgn	Configured Signal/Noise Margin for rate upshift. If the noise margin rises above this level, the modem should attempt to increase its transmit rate. Default: 90
minDownshiftTime	Minimum time in seconds that the current margin is below downshiftSnrMgn before a downshift occurs. Default: 30
minUpshiftTime	Minimum time in seconds that the current margin is above upshiftSnrMgn before an upshift occurs. Default: 30
bitSwap	Enable or disable downstream bit swap. Default: enabled
minINP	The minimum impulse noise protection for the downstream bearer channel expressed in symbols. One symbol equals 250 uS. noProtection, halfSymbol, singleSymbol, twoSymbols, threeSymbols, fourSymbols, fiveSymbols, sixSymbols, sevenSymbols, eightSymbols, nineSymbols, tenSymbols, elevenSymbols, twelveSymbols, thirteenSymbols, fourteenSymbols, fifteenSymbols, sixteenSymbols Default: twoSymbols
maxInterleaveDelay	Specifies the maximum interleave delay for the downstream slow channel. Default: 20
phyRSupport	Enable or disable downstream PHYR. enable disable Default: enable

Table 114: vdsl-co-config parameter definitions (Continued)

Parameter	Description
phyRmaxINP	PHYR maximum downstream impulse noise protection. A value of 0 specifies no protection. The values 5 through 160 specify the number of symbols in 1/10 increments. Default: 0
phyRminRSoverhead	PHYR minimum downstream RS overhead. Default: 0
phyRRtxRatio	PHYR minimum downstream fraction of the line rate allocated for retransmission. Default: 0
ginpVdslCoSupport	Enable or disable downstream G.INP / ITU-G.998.4. Only supported by Broadcom ports. 1 enable 2 disable
ginpVdslCoEtrMax	Maximum allowed value for downstream expected throughput (ETR) in kbit/s. The valid values are all multiples of 8 from 0 to the maximum of the valid values of the maximum net data rate specified in the associated Recommendation. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 100000 kbps
ginpVdslCoEtrMin	Minimum allowed value for downstream expected throughput (ETR) in kbit/s. The valid values are all multiples of 8 from 0 to the maximum of the valid values of the minimum net data rate specified the associated Recommendation. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 64
ginpVdslCoNdrMax	Maximum allowed value for downstream net data rate (NDR) in kbit/s. The valid values are all multiples of 8 from 0 to the maximum of the valid values of the maximum net data rate specified in the associated Recommendation. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 100000 kbps
ginpVdslCoShineRatio	The downstream loss of rate in a 1 second interval expressed as a fraction of NDR due to a single high impulse noise event (SHINE) impulse noise environment expected by the operator to occur at a pro ginpVdslCpeEtrMin ability acceptable for the services. The valid values are all multiples of 0.001 from 0 to 0.1. This field uses 1 to equal 0.001 and 100 to equal 0.1. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 10

Table 114: vdsl-co-config parameter definitions (Continued)

Parameter	Description
ginpVdslCoLeftrThreshold	The downstream rate Threshold below which the Low Error Free Rate (LEFTR) defect is declared. The threshold used to declare leftr defects is expressed in fraction of the net data rate (NDR). The value 0 is a special value to indicate that the receiver shall use a special value for declaring leftr defect. The minimum valid threshold to declare leftr is ETR/2. The receiver shall ignore threshold values that are less than the minimum and shall use ETR/2 for declaring leftr defect instead. The valid values are all multiples of 0.01 from 0.01 to 0.99. This field uses 1 to equal 0.01 and 99 to equal 0.99. Default: 0
ginpVdslCoMaxDelay	The maximum downstream delay in ms. This is the upper limit for the delay that is added to the transmission delay only caused by retransmissions. Here the receiver and/or the transmitter shall identify and discard all DTUs whose payload cannot be transferred over the reference point at the receiver without violating the delay_max limit. The time stamp shall be the criterion for discarding the DTUs. The processing delay between the U-interface and the retransmission sub-layer of the receiver in the retransmission data path direction shall be excluded from consideration for delay_max in the retransmission data path direction. The valid values are all integers from 1 to 63. ITU-T G.998.4 7.1.1 Control parameters, 7.1.2 Valid configurations, and 8.1.6 Time Stamp. Default: 20 mSecs
ginpVdslCoMinDelay	The minimum downstream delay in ms. This is the lower limit for the delay that is added to the transmission delay caused by retransmissions only. The time stamp shall be used by the outlet shaping function to determine when the payload of the DTU shall be sent to the reference point to meet the delay limits. The outlet shaping function shall minimize the additional delay that may be introduced above delay_min, and shall never exceed delay_max. The valid values are all integers from 0 to 63. ITU-T G.998.4 7.1.1 Control parameters, 7.1.2 Valid configurations, and 8.1.6 Time Stamp. Default: 0
ginpVdslCoMin	The minimum downstream impulse noise protection (INP) against single high impulse noise event (SHINE) in discrete multitone (DMT) symbols. The valid values are all integers from 0 to 63 for system with a sub-carrier spacing of 4.3125 kHz. The valid values are all integers from 0 to 127 for system with a sub-carrier spacing of 8.625 kHz. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 4
ginpVdslCoMinRSoverhead	This value specifies the downstream bandwidth reserved for RS (reed-solomon) codewords. The minimum guaranteed R/N ratio. The unit is 1/256th and the range is 0..64 (0 to 25%). Default: 0

Table 114: vdsl-co-config parameter definitions (Continued)

Parameter	Description
ginpVdslCoReinCfg	The minimum downstream impulse protection against electrical repetitive impulse noise (REIN) in DMT symbols. The valid values are all integers from 0 to 7 for system with a sub-carrier spacing of 4.3125 kHz. The valid values are all integers from 0 to 13 for system with a sub-carrier spacing of 8.625 kHz. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 0
ginpVdslCoReinFreq	Specifies the frequency of REIN inter-arrival time. It is used in the Channel Initialization Policy and on-line reconfiguration procedures. REIN is commonly coupled from electrical power cables appliances drawing power from the AC electrical power network, having a repetition rate of twice the AC power frequency (100 or 120 Hz). The valid values are integers 100 hz or 120 hz. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations." 1 freq100hz 2 freq120hz Default: freq120hz
ginpVdslCoRtxMode	Downstream retransmission Mode (RTX MODE). The RTX_MODE is a configuration parameter used to control activation of retransmission during initialization. This parameter has 4 valid values: FORBIDDEN: ITU-T G.998.4 retransmission not allowed. PREFERRED: ITU-T G.998.4 retransmission is preferred by the operator. (i.e., if ITU-T G.998.4 RTX capability is supported by both XTU's, the XTU's shall select ITU-T G.998.4 operation for this direction). FORCED: Force the use of the ITU-T G.998.4 retransmission.(i.e., if ITU-T G.998.4 RTX capability in this direction is not supported by both XTU's or not selected by the XTU's, an initialization failure shall result). NOTE: Due to the optionality of ITU-T G.998.4 retransmission in upstream direction, the use of FORCED in upstream may lead to initialization failure, even if the XTU is supporting ITU-T G.998.4 (in downstream). TESTMODE: Force the use of the ITU-T G.998.4 retransmission in the test mode described in clause 10.4. (i.e., if ITU-T G.998.4 RTX capability is not supported by both XTU's or not selected by the XTU's, an initialization failure shall result).ITU-T G.998.4 11.1.13 Retransmission Mode (RTX_MODE). 1 forbidden 2 preferred 3 forced 4 testmode Default: preferred

View vdsl-cpe-config profile default parameters

The VDSL2 upstream interface is the **vdsl-cpe-config** profile which defines upstream behavior.

The Broadcom Phy-R™ parameter **phyRSupport** in the **vdsl-cpe-config** profile is enabled by default. The **line-type** parameter in the **vdsl-config** profile must be set to *interleavedonly* for Phy-R™ to work. See [Modifying the default vdsl-config parameters on page 1033](#).

Viewing vdsl-cpe-config profile defaults

View the **vdsl-cpe-config** parameters and their default values.

1 Verify the VDSL2 cards.

```
zSH> slots
MXK 819
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
  1: MXK 24 PORT VDSL2 (RUNNING)
  5: MXK ADSL-48-B Bonded (RUNNING)
  6: MXK ADSL-48-B Bonded (RUNNING)
```

2 View the vdsl-cpe-config parameters and values.

```
zSH> show vdsl-cpe-config
fastMaxTxRate:-----> {0 - 60000}
fastMinTxRate:-----> {0 - 60000}
interleaveMaxTxRate:-----> {0 - 60000}
interleaveMinTxRate:-----> {0 - 60000}
rateMode:-----> manual adapt-at-init dynamic
maxPower:-----> {-130 - 200}
maxSnrMgn:-----> {0 - 310}
minSnrMgn:-----> {0 - 310}
targetSnrMgn:-----> {0 - 310}
pbo-control:-----> disabled auto
pbo-psd-template:-----> ansi-a ansi-f etsi-a etsi-b etsi-c etsi-d
etsi-e etsi-f ab-param
pbo-psd-param-a1:-----> {4000 - 8096}
pbo-psd-param-a2:-----> {4000 - 8096}
pbo-psd-param-b1:-----> {0 - 4096}
pbo-psd-param-b2:-----> {0 - 4096}
downshiftSnrMgn:-----> {0 - 310}
upshiftSnrMgn:-----> {0 - 310}
minDownshiftTime:-----> {0 - 16383}
minUpshiftTime:-----> {0 - 16383}
bitSwap:-----> disabled enabled
minINP:-----> noprotection halvesymbol singlesymbol
twosymbols threesymbols foursymbols fivesymbols sixsymbols sevensymbols
eightsymbols ninesymbols tensymbols elevensymbols twelvesymbols
thirteensymbols fourteensymbols fifteensymbols sixteensymbols
```

```

maxInterleaveDelay:-----> {0 - 63}
phyRSupport:-----> enable disable
phyRmaxINP:-----> {0 - 160}
phyRminRSoverhead:-----> {0 - 255}
phyRRtxRatio:-----> {0 - 255}
pbo-psd-param-a3:-----> {4000 - 8096}
pbo-psd-param-a4:-----> {4000 - 8096}
pbo-psd-param-b3:-----> {0 - 4096}
pbo-psd-param-b4:-----> {0 - 4096}
ginpVdslCpeSupport:-----> enable disable
ginpVdslCpeEtrMax:-----> {64 - 2000000}
ginpVdslCpeEtrMin:-----> {64 - 200000}
ginpVdslCpeNdrMax:-----> {64 - 200000}
ginpVdslCpeShineRatio:-----> {0 - 255}
ginpVdslCpeLeftrThreshold:-----> {0 - 99}
ginpVdslCpeMaxDelay:-----> {0 - 63}
ginpVdslCpeMinDelay:-----> {0 - 63}
ginpVdslCpeMin:-----> {0 - 127}
ginpVdslCpeMinRSoverhead:-----> {0 - 64}
ginpVdslCpeReinCfg:-----> {0 - 7}
ginpVdslCpeReinFreq:-----> freq100hz freq120hz
ginpVdslCpeRtxMode:-----> forbidden preferred forced testmode

```

3 View the current **vdsl-cpe-config** default parameters for a VDSL2 port.

```

zSH> get vdsl-cpe-config 1-2-1-0/vdsl
vdsl-cpe-config 1-2-1-0/vdsl
fastMaxTxRate: -----> {60000}
fastMinTxRate: -----> {0}
interleaveMaxTxRate: -----> {60000}
interleaveMinTxRate: -----> {0}
rateMode: -----> {dynamic}
maxPower: -----> {145}
maxSnrMgn: -----> {160}
minSnrMgn: -----> {0}
targetSnrMgn: -----> {60}
pbo-control: -----> {disabled}
pbo-psd-template: -----> {ansi-a}
pbo-psd-param-a1: -----> {4000}
pbo-psd-param-a2: -----> {4000}
pbo-psd-param-b1: -----> {4000}
pbo-psd-param-b2: -----> {4000}
downshiftSnrMgn: -----> {30}
upshiftSnrMgn: -----> {90}
minDownshiftTime: -----> {30}
minUpshiftTime: -----> {30}
bitSwap: -----> {enabled}
minINP: -----> {twosymbols}
maxInterleaveDelay: -----> {20}
phyRSupport: -----> {enable}
phyRmaxINP: -----> {0}
phyRminRSoverhead: -----> {0}
phyRRtxRatio: -----> {0}
pbo-psd-param-a3: -----> {4000}

```

```

pbo-psd-param-a4: -----> {4000}
pbo-psd-param-b3: -----> {4000}
pbo-psd-param-b4: -----> {4000}
ginpVdslCpeSupport: -----> {disable}
ginpVdslCpeEtrMax: -----> {60000}
ginpVdslCpeEtrMin: -----> {64}
ginpVdslCpeNdrMax: -----> {60000}
ginpVdslCpeShineRatio: -----> {10}
ginpVdslCpeLeftrThreshold: ----> {0}
ginpVdslCpeMaxDelay: -----> {20}
ginpVdslCpeMinDelay: -----> {0}
ginpVdslCpeMin: -----> {4}
ginpVdslCpeMinRSoverhead: ----> {0}
ginpVdslCpeReinCfg: -----> {0}
ginpVdslCpeReinFreq: -----> {freq120hz}
ginpVdslCpeRtxMode: -----> {preferred}

```

Table 115 defines the parameter values for the **vdsl-cpe-config** profile.

Table 115: vdsl-cpe-config parameter definitions

Parameter	Definition
fastMaxTxRate	Specifies the maximum upstream fast channel data rate in steps of 1000 bits/second. The maximum aggregate upstream transmit speed of the line can be derived from the sum of maximum upstream fast and slow channel data rates. Default: 200,000
fastMinTxRate	Specifies the minimum upstream fast channel data rate in steps of 1000 bits/second. The minimum aggregate upstream transmit speed of the line can be derived from the sum of minimum upstream fast and slow channel data rates. Default: 0
interleaveMaxTxRate	Specifies the maximum upstream slow channel data rate in steps of 1000 bits/second. Default: 200,000
interleaveMinTxRate	Specifies the minimum upstream slow channel data rate in steps of 1000 bits/second. Default: 0
rateMode	Specifies the rate selection behavior for the line in the upstream direction. manual: forces the rate to the configured rate adapt-at-init: adapts the line at initialization only dynamic: adapts the line at initialization and showtime Default: dynamic
maxPower	Specifies the maximum aggregate upstream power level in the range 0 to 14.5 dBm. Default: 130

Table 115: vdsi-cpe-config parameter definitions (Continued)

Parameter	Definition
maxSnrMgn	Specifies the maximum upstream Signal/Noise Margin in units of 0.10 dB, for a range of 0 to 31.0 dB. Default: 127
minSnrMgn	Specifies the minimum upstream Signal/Noise Margin in units of 0.10 dB, for a range of 0 to 31.0 dB. Default: 0
targetSnrMgn	Specifies the target upstream Signal/Noise Margin in units of 0.10 dB, for a range of 0 to 31.0 dB. This is the Noise Margin the transceivers must achieve with a BER of 10^{-7} or better to successfully complete initialization. Default: 60
downshiftSnrMgn	Configured Signal/Noise Margin for rate downshift. If the noise margin falls below this level, the modem should attempt to decrease its transmit rate. Default: 30
upshiftSnrMgn	Configured Signal/Noise Margin for rate upshift. If the noise margin rises above this level, the modem should attempt to increase its transmit rate. Default: 90
minDownshiftTime	Minimum time that the current margin is below DownshiftSnrMgn before a downshift occurs. Default: 30
minUpshiftTime	Minimum time that the current margin is above UpshiftSnrMgn before an upshift occurs. Default: 30
bitSwap	Enable or disable upstream bit swap. enabled disabled Default: enabled
minINP	The minimum impulse noise protection for the upstream bearer channel expressed in symbols. One symbol equals 250 uS. noProtection, halfSymbol, singleSymbol, twoSymbols, fourSymbols, eightSymbols, sixteenSymbols Default: twoSymbols
maxInterleaveDelay	The maximum interleaving delay in the upstream direction. A value of zero indicates no delay introduced. Default: 20
phyRSupport	Enable or disable upstream PHYR. Default: enable

Table 115: vdsl-cpe-config parameter definitions (Continued)

Parameter	Definition
phyRmaxINP	PHYR maximum upstream impulse noise protection. A value of 0 specifies no protection. The values 5 through 160 specify the number of symbols in 1/10 increments. Default: 0
phyRminRSoverhead	PHYR minimum upstream RS overhead. Default: 0
phyRRtxRatio	PHYR minimum upstream fraction of the line rate allocated for retransmission. Default: 0
pbo-psd-param-a3	Upstream power backoff PSD parameter A3. This parameter is only used if the mask selection is set to PSDMaskABParameters. Default: 4000
pbo-psd-param-a4	Upstream power backoff PSD parameter A4. This parameter is only used if the mask selection is set to PSDMaskABParameters. Default: 4000
pbo-psd-param-b3	Upstream power backoff PSD parameter B3. This parameter is only used if the mask selection is set to PSDMaskABParameters. Default: 4000
pbo-psd-param-b4	Upstream power backoff PSD parameter B4. This parameter is only used if the mask selection is set to PSDMaskABParameters. Default: 4000
ginpVdslCpeSupport	1 enable 2 disable Enable or disable upstream G.INP / ITU-G.998.4. Only supported by Broadcom ports.
ginpVdslCpeEtrMax	Maximum allowed value for upstream expected throughput (ETR) in kbit/s. The valid values are all multiples of 8 from 0 to the maximum of the valid values of the maximum net data rate specified in the associated Recommendation. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. REFERENCE ITU-G.998.4 Default: 60000 kbps
ginpVdslCpeEtrMin	Minimum allowed value for upstream expected throughput (ETR) in kbit/s. The valid values are all multiples of 8 from 0 to the maximum of the valid values of the minimum net data rate specified in the associated Recommendation. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 64 kbps

Table 115: vdsl-cpe-config parameter definitions (Continued)

Parameter	Definition
ginpVdslCpeNdrMax	Maximum allowed value for upstream net data rate (NDR) in kbit/s. The valid values are all multiples of 8 from 0 to the maximum of the valid values of the maximum net data rate specified in the associated Recommendation. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 60000 kbps
ginpVdslCpeShineRatio	The upstream loss of rate in a 1 second interval expressed as a fraction of NDR due to a single high impulse noise event (SHINE) impulse noise environment expected by the operator to occur at a probability acceptable for the services. The valid values are all multiples of 0.001 from 0 to 0.1. This field uses 1 to equal 0.001 and 100 to equal 0.1. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 10
ginpVdslCpeLeftrThreshold	The upstream rate Threshold below which the Low Error Free Rate (LEFTR) defect is declared. The threshold used to declare leftr defects is expressed in fraction of the net data rate (NDR). The value 0 is a special value to indicate that the receiver shall use a special value for declaring leftr defect. The minimum valid threshold to declare leftr is ETR/2. The receiver shall ignore threshold values that are less than the minimum and shall use ETR/2 for declaring leftr defect instead. The valid values are all multiples of 0.01 from 0.01 to 0.99. This field uses 1 to equal 0.01 and 99 to equal 0.99. Default: 0
ginpVdslCpeMaxDelay	The maximum upstream delay in ms. This is the upper limit for the delay that is added to the transmission delay only caused by retransmissions. Here the receiver and/or the transmitter shall identify and discard all DTUs whose payload cannot be transferred over the reference point at the receiver without violating the delay_max limit. The time stamp shall be the criterion for discarding the DTUs. The processing delay between the U-interface and the retransmission sub-layer of the receiver in the retransmission data path direction shall be excluded from consideration for delay_max in the retransmission data path direction. The valid values are all integers from 1 to 63. ITU-T G.998.4 7.1.1 Control parameters, 7.1.2 Valid configurations, and 8.1.6 Time Stamp. Default: 20
ginpVdslCpeMinDelay	The minimum upstream delay in ms. This is the lower limit for the delay that is added to the transmission delay caused by retransmissions only. The time stamp shall be used by the outlet shaping function to determine when the payload of the DTU shall be sent to the reference point to meet the delay limits. The outlet shaping function shall minimize the additional delay that may be introduced above delay_min, and shall never exceed delay_max. The valid values are all integers from 0 to 63. ITU-T G.998.4 7.1.1 Control parameters, 7.1.2 Valid configurations, and 8.1.6 Time Stamp. Default: 0

Table 115: vdsl-cpe-config parameter definitions (Continued)

Parameter	Definition
ginpVdslCpeMin	The minimum upstream impulse noise protection (INP) against single high impulse noise event (SHINE) in discrete multitone (DMT) symbols. The valid values are all integers from 0 to 63 for system with a sub-carrier spacing of 4.3125 kHz. The valid values are all integers from 0 to 127 for system with a sub-carrier spacing of 8.625 kHz. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 4
ginpVdslCpeMinRSoverhead	This value specifies the upstream bandwidth reserved for RS (reed-solomon) codewords. The minimum guaranteed R/N ratio. The unit is 1/256th and the range is 0..64 (0 to 25%). Default: 0
ginpVdslCpeReinCfg	The minimum upstream impulse protection against electrical repetitive impulse noise (REIN) in DMT symbols. The valid values are all integers from 0 to 7 for system with a sub-carrier spacing of 4.3125 kHz. The valid values are all integers from 0 to 13 for system with a sub-carrier spacing of 8.625 kHz. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 0
ginpVdslCpeReinFreq	Specifies the frequency of REIN inter-arrival time. It is used in the Channel Initialization Policy and on-line reconfiguration procedures. REIN is commonly coupled from electrical power cables appliances drawing power from the AC electrical power network, having a repetition rate of twice the AC power frequency (100 or 120 Hz). The valid values are integers 100 hz or 120 hz. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. 1 freq100hz 2 freq120hz Default: freq120hz

Table 115: vdsl-cpe-config parameter definitions (Continued)

Parameter	Definition
ginpVdslCpeRtxMode	Upstream retransmission Mode (RTX MODE). The RTX_MODE is a configuration parameter used to control activation of retransmission during initialization. This parameter has 4 valid values: FORBIDDEN: ITU-T G.998.4 retransmission not allowed. PREFERRED: ITU-T G.998.4 retransmission is preferred by the operator. (i.e., if ITU-T G.998.4 RTX capability is supported by both XTU's, the XTU's shall select ITU-T G.998.4 operation for this direction). FORCED: Force the use of the ITU-T G.998.4 retransmission. (i.e., if ITU-T G.998.4 RTX capability in this direction is not supported by both XTU's or not selected by the XTU's, an initialization failure shall result). NOTE: Due to the optionality of ITU-T G.998.4 retransmission in upstream direction, the use of FORCED in upstream may lead to initialization failure, even if the XTU is supporting ITU-T G.998.4 (in downstream). TESTMODE: Force the use of the ITU-T G.998.4 retransmission in the test mode described in clause 10.4. (i.e., if ITU-T G.998.4 RTX capability is not supported by both XTU's or not selected by the XTU's, an initialization failure shall result). ITU-T G.998.4 11.1.13 Retransmission Mode (RTX_MODE) REFERENCE ITU-G.998.4. 1 forbidden 2 preferred 3 forced 4 testmode Default: perferred

Configure VDSL2 profiles to cap train rates

You can control upstream and downstream train rates in Kbps for *fast* or *interleaved* modes in the **vdsl-config**, **vdsl-co-config**, and **vdsl-cpe-config** profiles.

[Table 116](#) shows the profiles and default parameters for upstream and downstream train rates.

Table 116: Profiles and parameters for capping upstream and downstream train rates

Profile	Parameter and train rates
vdsl-config	line-type: <i>fastonly</i> or <i>interleaveonly</i>

Table 116: Profiles and parameters for capping upstream and downstream train rates (Continued)

Profile	Parameter and train rates
vdsl-co-config	fastMaxTxRate: {200000} fastMinTxRate: {0} or interleaveMaxTxRate: {200000} interleaveMinTxRate: {0}
vdsl-cpe-config	fastMaxTxRate: {200000} fastMinTxRate: {0} or interleaveMaxTxRate: {200000} interleaveMinTxRate: {0}

Configure VDSL2 G.INP

G.INP is a standards based error correction mechanism replacing Phy-R™.



Note: G.INP provides retransmission service on VDSL2 upstream and downstream and on ADSL2+ downstream only.



Note: The G.INP standard does not cover ADSL, and as such, G.INP on ADSL is not supported.

Enabling G.INP

Enable the G.INP support parameter in both the **vdsl-co-config** profile and the **vdsl-cpe-config** profile.

- 1 Update the **ginpVdslCoSupport** parameter in the **vdsl-co-config** profile to enable G.INP.

```

zSH> update vdsl-co-config 1-2-1-0/vdsl
vdsl-co-config 1-2-1-0/vdsl
Please provide the following: [q]uit.
fastMaxTxRate: -----> {100000}:
fastMinTxRate: -----> {0}:
interleaveMaxTxRate: -----> {100000}:
interleaveMinTxRate: -----> {0}:
rateMode: -----> {dynamic}:
maxPower: -----> {145}:
maxSnrMgn: -----> {160}:
minSnrMgn: -----> {0}:
targetSnrMgn: -----> {60}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minDownshiftTime: -----> {30}:
minUpshiftTime: -----> {30}:

```

```

bitSwap: -----> {enabled}:
minINP: -----> {twosymbols}:
maxInterleaveDelay: -----> {20}:
phyRSupport: -----> {enable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
ginpVdslCoSupport: -----> {disable}: enable
ginpVdslCoEtrMax: -----> {100000}:
ginpVdslCoEtrMin: -----> {64}:
ginpVdslCoNdrMax: -----> {100000}:
ginpVdslCoShineRatio: -----> {10}:
ginpVdslCoLeftrThreshold: -----> {0}:
ginpVdslCoMaxDelay: -----> {20}:
ginpVdslCoMinDelay: -----> {0}:
ginpVdslCoMin: -----> {4}:
ginpVdslCoMinRSoverhead: -----> {0}:
ginpVdslCoReinCfg: -----> {0}:
ginpVdslCoReinFreq: -----> {freq120hz}:
ginpVdslCoRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

2 Update the **ginpVdslCpeSupport** parameter in the **vdsl-cpe-config** profile to enable G.INP.

```

zSH> update vdsl-cpe-config 1-2-1-0/vdsl
vdsl-cpe-config 1-2-1-0/vdsl
Please provide the following: [q]uit.
fastMaxTxRate: -----> {60000}:
fastMinTxRate: -----> {0}:
interleaveMaxTxRate: -----> {60000}:
interleaveMinTxRate: -----> {0}:
rateMode: -----> {dynamic}:
maxPower: -----> {145}:
maxSnrMgn: -----> {160}:
minSnrMgn: -----> {0}:
targetSnrMgn: -----> {60}:
pbo-control: -----> {disabled}:
pbo-psd-template: -----> {ansi-a}:
pbo-psd-param-a1: -----> {4000}:
pbo-psd-param-a2: -----> {4000}:
pbo-psd-param-b1: -----> {4000}:
pbo-psd-param-b2: -----> {4000}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minDownshiftTime: -----> {30}:
minUpshiftTime: -----> {30}:
bitSwap: -----> {enabled}:
minINP: -----> {twosymbols}:
maxInterleaveDelay: -----> {20}:
phyRSupport: -----> {enable}:
phyRmaxINP: -----> {0}:

```

```

phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
pbo-psd-param-a3: -----> {4000}:
pbo-psd-param-a4: -----> {4000}:
pbo-psd-param-b3: -----> {4000}:
pbo-psd-param-b4: -----> {4000}:
ginpVdslCpeSupport: -----> {disable}: enable
ginpVdslCpeEtrMax: -----> {60000}:
ginpVdslCpeEtrMin: -----> {64}:
ginpVdslCpeNdrMax: -----> {60000}:
ginpVdslCpeShineRatio: -----> {10}:
ginpVdslCpeLeftrThreshold: ----> {0}:
ginpVdslCpeMaxDelay: -----> {20}:
ginpVdslCpeMinDelay: -----> {0}:
ginpVdslCpeMin: -----> {4}:
ginpVdslCpeMinRSoverhead: -----> {0}:
ginpVdslCpeReinCfg: -----> {0}:
ginpVdslCpeReinFreq: -----> {freq120hz}:
ginpVdslCpeRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

VDSL2 statistics

This chapter describes:

- [View VDSL2 statistics, page 1116](#)
- [View VDSL2 statistics for vectoring, page 1053](#)
- [View VDSL2 statistics with the -v variable, page 1117](#)
- [Clear VDSL2 counters, page 1119](#)
- [VDSL statistics parameters, page 1119](#)

View VDSL2 statistics

Use the **dslstat** *shelf/slot/port/subport/interface type* command to retrieve or clear DSL statistics for any VDSL port in the system. Entering the **dslstat** command with the -v (verbose) variable retrieves all available statistics.

Viewing VDSL2 statistics

View VDSL2 statistics.

```

zSH> dslstat 1-11-1-0/vdsl1
General Stats:
-----
AdminStatus.....UP
LineStatus.....DATA
Line uptime (DD:HH:MM:SS).....0:00:21:39
DslUpLineRate (bitsPerSec).....23628000

```

```

DslDownLineRate (bitsPerSec).....65239000
DslMaxAttainableUpLineRate (bitsPerSec).....23677000
DslMaxAttainableDownLineRate (bitsPerSec)....64496000
In Octets.....0
In Pkts/Cells/Frags.....84341
In Discards.....0
In Errors.....0
Out Octets.....0
Out Pkts/Cells/Frags.....0
Out Discards.....0
Out Errors.....0
DSL Physical Stats:
-----
Actual Transmission connection standard.....VDSL2
Vdsl2CurrentProfile.....g993-2-17a
DslLineSnrMgn (tenths dB).....61
DslLineAtn (tenths dB).....0
DslCurrOutputPwr (tenths dB).....145
LOFS.....0
LOLS.....60
LOSS.....60
ESS.....60
CRC Errors.....0
Inits.....2
near-end statistics:
-----
Loss of Frame Seconds.....0
Loss of Signal Seconds.....60
Loss of Link Seconds.....60
Severely Errored Seconds.....60
Unavailable Seconds.....60
far-end statistics:
-----
Loss of Frame Seconds.....0
Loss of Signal Seconds.....60
Loss of Link Seconds.....0
Severely Errored Seconds.....60
Unavailable Seconds.....60
Loss of Power Seconds (LPRS).....0
phyR Statistics:
-----
Vtuc PhyRActive.....FALSE
Vtuc Retransmitted codewords.....0
Vtuc Corrected Retransmitted codewords.....0
Vtuc UnCorrectable Retransmitted codewords...0
Vtur PhyRActive.....FALSE
Vtur Retransmitted codewords.....0
Vtur Corrected Retransmitted codewords.....0
Vtur UnCorrectable Retransmitted codewords...0
G.INP Statistics:
-----
Vtuc ginpActive.....TRUE
Vtuc Error Free Throughput Rate (LEFTR) Secs.0
Vtuc Error Free Bits.....467640

```



```

Vtuc Minimum Error Free Throughput Rate.....23621
Vtur ginpActive.....TRUE
Vtur Error Free Throughput Rate (LEFTR) Secs.0
Vtur Error Free Bits.....1292505
Vtur Minimum Error Free Throughput Rate.....65218

```

View VDSL2 statistics for vectoring

Use the **dslstat** command on VDSL2 cards with vectoring for vectoring statistics.

```
zSH> dslstat 1-4-1-0/vdsl1
```

General Stats:

```

AdminStatus.....UP
LineStatus.....DATA
Line uptime (DD:HH:MM:SS).....0:00:15:57
DslUpLineRate (bitsPerSec).....46787000
DslDownLineRate (bitsPerSec).....99976000
DslMaxAttainableUpLineRate (bitsPerSec).....44817000
DslMaxAttainableDownLineRate (bitsPerSec)....144504000
In Octets.....0
In Pkts/Cells/Frags.....2865
In Discards.....0
In Errors.....0
Out Octets.....0
Out Pkts/Cells/Frags.....0
Out Discards.....0
Out Errors.....0

```

DSL Physical Stats:

```

Actual Transmission connection standard.....VDSL2 / Vectoring
Vdsl2CurrentProfile.....g993-2-17a
DslLineSnrMgn (tenths dB).....91
DslLineAtn (tenths dB).....0
DslCurrOutputPwr (tenths dB).....122
LOFS.....0
LOLS.....107
LOSS.....107
ESS.....108
CRC Errors.....1
Inits.....1

```

DSL Vectoring Stats:

```

EsDsCounter.....86
EsUsCounter.....1071
IsDsFeValid.....1
EsDsFeCounter.....34236

```

near-end statistics:

```

Loss of Frame Seconds.....0
Loss of Signal Seconds.....107
Loss of Link Seconds.....107
Severely Errored Seconds.....107

```

```

Unavailable Seconds.....107
far-end statistics:
-----
Loss of Frame Seconds.....0
Loss of Signal Seconds.....107
Loss of Link Seconds.....0
Severely Errored Seconds.....107
Unavailable Seconds.....107
Loss of Power Seconds (LPRS).....0
phyR Statistics:
-----
Vtuc PhyRActive.....FALSE
Vtuc Retransmitted codewords.....0
Vtuc Corrected Retransmitted codewords.....0
Vtuc UnCorrectable Retransmitted codewords...0
Vtur PhyRActive.....FALSE
Vtur Retransmitted codewords.....0
Vtur Corrected Retransmitted codewords.....0
Vtur UnCorrectable Retransmitted codewords...0
G.INP Statistics:
-----
Vtuc ginpActive.....FALSE
Vtuc Error Free Throughput Rate (LEFTR) Secs.0
Vtuc Error Free Bits.....0
Vtuc Minimum Error Free Throughput Rate.....0
Vtur ginpActive.....FALSE
Vtur Error Free Throughput Rate (LEFTR) Secs.0
Vtur Error Free Bits.....0
Vtur Minimum Error Free Throughput Rate.....0

```

View VDSL2 statistics with the -v variable

Using the -v (verbose) variable with the **dslstat** command displays all available statistics.



Note: Statistics in bold indicate Phy-R™ statistics.

View all available VDSL2 statistics

To view all available statistics on VDSL2 enter:

```

zSH> dslstat 1-11-1-0/vdsl1 -v
General Stats:
-----
AdminStatus.....UP
LineStatus.....DATA
Line uptime (DD:HH:MM:SS).....0:00:26:51
DslUpLineRate (bitsPerSec).....23628000
DslDownLineRate (bitsPerSec).....65239000
DslMaxAttainableUpLineRate (bitsPerSec).....23685000
DslMaxAttainableDownLineRate (bitsPerSec)....64496000

```

```

In Octets.....0
In Pkts/Cells/Frags.....84592
In Discards.....0
In Errors.....0
Out Octets.....0
Out Pkts/Cells/Frags.....0
Out Discards.....0
Out Errors.....0
DSL Physical Stats:
-----
Actual Transmission connection standard.....VDSL2
Vdsl2CurrentProfile.....g993-2-17a
DslLineSnrMgn (tenths dB).....61
DslLineAtn (tenths dB).....0
DslCurrOutputPwr (tenths dB).....145
LOFS.....0
LOLS.....60
LOSS.....60
ESS.....60
CRC Errors.....0
Inits.....2
near-end statistics:
-----
Loss of Frame Seconds.....0
Loss of Signal Seconds.....60
Loss of Link Seconds.....60
Severely Errored Seconds.....60
Unavailable Seconds.....60
far-end statistics:
-----
Loss of Frame Seconds.....0
Loss of Signal Seconds.....60
Loss of Link Seconds.....0
Severely Errored Seconds.....60
Unavailable Seconds.....60
Loss of Power Seconds (LPRS).....0
phyR Statistics:
-----
Vtuc PhyRActive.....FALSE
Vtuc Retransmitted codewords.....0
Vtuc Corrected Retransmitted codewords.....0
Vtuc UnCorrectable Retransmitted codewords...0
Vtur PhyRActive.....FALSE
Vtur Retransmitted codewords.....0
Vtur Corrected Retransmitted codewords.....0
Vtur UnCorrectable Retransmitted codewords...0
G.INP Statistics:
-----
Vtuc ginpActive.....TRUE
Vtuc Error Free Throughput Rate (LEFTR) Secs.0
Vtuc Error Free Bits.....579960
Vtuc Minimum Error Free Throughput Rate.....23621
Vtur ginpActive.....TRUE
Vtur Error Free Throughput Rate (LEFTR) Secs.0

```

```

Vtur Error Free Bits.....1602945
Vtur Minimum Error Free Throughput Rate.....65218
XTUC PHY Stats:
-----
serialNumber.....121 v10.08.27, 2012-10-24
vendorId.....BDCM 0x4d54
versionNumber.....VE_10_8_27
curSnrMargin (tenths dB).....61
currAtn (tenths dB).....0
currStatus.....NO DEFECT
currOutputPwr (tenths dB).....145
currAttainableRate (bitsPerSec).....64496000
currLineRate (bitsPerSec).....65239000
XTUC CHAN Stats:
-----
interleaveDelay (tenths milliseconds).....0
crcBlockLength (bytes).....0
currTxRate (bitsPerSec).....65239000
currTxSlowBurstProt.....0
currTxFastFec.....0
XTUR PHY Stats:
-----
serialNumber.....300142437
vendorId.....BDCM 0
versionNumber.....A2pv6C038h
curSnrMargin (tenths dB).....61
currAtn (tenths dB).....25
currStatus.....NO DEFECT
currOutputPwr (tenths dB).....43
currAttainableRate (bitsPerSec).....23685000
currLineRate (bitsPerSec).....23628000
XTUR CHAN Stats:
-----
interleaveDelay (tenths milliseconds).....0
crcBlockLength (bytes).....0
currTxRate (bitsPerSec).....23628000
currTxSlowBurstProt.....0
currTxFastFec.....0

```

Clear VDSL2 counters

Clearing DSL counters

You can clear DSL counters to make identifying the changing statistics easier to read.

- 1 Clear the statistics using the **dslstat clear** command

```
zSH> dslstat clear 1-11-1-0/vdsl1
```

- 2 View the changes.

For reference the **dslstat** command (see [Viewing VDSL2 statistics on page 1116](#)) shows the statistics prior to clearing the statistics. Statistics which are cleared by the **dslstat clear** command are in *bold*.

```
zSH> dslstat 1-11-1-0/vdsl
General Stats:
-----
AdminStatus.....UP
LineStatus.....DATA
Line uptime (DD:HH:MM:SS).....0:00:40:42
DslUpLineRate (bitsPerSec).....23579000
DslDownLineRate (bitsPerSec).....65556000
DslMaxAttainableUpLineRate (bitsPerSec).....23625000
DslMaxAttainableDownLineRate (bitsPerSec)....64804000
In Octets.....0
In Pkts/Cells/Frags.....1990
In Discards.....0
In Errors.....0
Out Octets.....0
Out Pkts/Cells/Frags.....0
Out Discards.....0
Out Errors.....0
DSL Physical Stats:
-----
Actual Transmission connection standard.....VDSL2
Vdsl2CurrentProfile.....g993-2-17a
DslLineSnrMgn (tenths dB).....61
DslLineAtn (tenths dB).....0
DslCurrOutputPwr (tenths dB).....145
LOFS.....0
LOLS.....23
LOSS.....23
ESS.....23
CRC Errors.....0
InitS.....1
```

VDSL statistics parameters

[Table 123](#) defines the statistics displayed in the **dslstat** command for an VDSL line.

Table 117: VDSL2 statistics

Statistic	Description
General statistics:	

Table 117: VDSL2 statistics (Continued)

Statistic	Description
AdminStatus	Administrative status of the port: Values: Up Interface is ready to pass packets. Down Interface is unable to pass packets. Testing Interface is in a special testing state and is unable to pass packets.
LineStatus	Line status provides information about the VDSL2 link. Values for a single VDSL2 line: ACT — the line currently has link and can pass traffic in both directions OOS — the line does not have link TRAFFIC DISABLE — The line currently has link but not underlying VDSL2 protocol; traffic will not pass.
Line uptime (DD:HH:MM:SS)	How long the interface has been up in dd hh mm (day, hour, minute, second) format.
DslUpLineRate (bitsPerSec)	Displays the DSL upstream (customer premise > central office) line rate on this interface.
DslDownLineRate (bitsPerSec)	Displays the DSL downstream (central office > customer premise) line rate on this interface.
DslMaxAttainableUpLineRate (bitsPerSec)	Displays the maximum line rate that can be supported on this line in the upstream direction.
DslMaxAttainableDownLineRate (bitsPerSec)	Displays the maximum line rate that can be supported on this line in the downstream direction.
In Octets	Number of received octets.
In Pkts/Cells/Frags	Number of received packets/cells/frags.
In Discards	Number of received discards.
In Errors	Number of receive errors.
Out Octets	Number of transmitted octets.
Out Pkts/Cells/Frags	Number of transmitted packets/cells/frags.
Out Discards	Number of transmission discards.
Out Errors	Number of transmission errors.
DSL2 Physical Stats:	
Actual Transmission connection standard	Indicates the maximum line rate that can be supported on this line in the downstream direction.
Vdsl2CurrentProfile	The VDSL2 standard to be used for the line.

Table 117: VDSL2 statistics (Continued)

Statistic	Description
DslLineSnrMgn (tenths dB)	DSL Line Signal to Noise Ratio Margin — The strength of the DSL signal relative to the noise on line.
DslLineAtn (tenths dB)	DSL Line Attenuation — Measure of the signal degradation between the VDSL2 port and the modem.
DslCurrOutputPwr (tenths dB)	Not currently used.
LOFS	Number of Loss of Frame Seconds.
LOLS	Number of Loss of Line Seconds.
LOSS	Number of Loss of Signal Seconds.
ESS	Number of errored seconds (the number of one-second intervals containing one or more CRC anomalies or one or more LoS or Sef defects) that has been reported in the current 15-minute interval.
CRC Errors	Cyclic Redundancy Check Errors — CRC checks for transmission errors. The CRC code is computed from the data in the message. If the data is altered the CRC computation will not be in agreement with the data.
Initis	Number of line initialization attempts, including both successful and failed attempts.
near-end (CO) statistics:	
Loss of Frame Seconds	Count of seconds during this interval that there was Loss of Framing.
Loss of Signal Seconds	Count of seconds during this interval that there was Loss of Signal.
Loss of Link Seconds	Count of seconds during this interval that there was Loss of Link.
Severely Errored Seconds	Count of Severely Errored Seconds during this interval.
Unavailable Seconds	Count of Unavailable Seconds during this interval.
Retransmitted codewords	Retransmitted Codewords.
Corrected Retransmitted codewords	Retransmitted corrected Codewords.
UnCorrectable Retransmitted codewords	Retransmitted uncorrectable Codewords.
far-end statistics:	
Loss of Frame Seconds	Count of seconds during this interval that there was Loss of Framing.
Loss of Signal Seconds	Count of seconds during this interval that there was Loss of Signal.
Loss of Link Seconds	Count of seconds during this interval that there was Loss of Link.
Severely Errored Seconds	Count of Severely Errored Seconds during this interval.
Unavailable Seconds	Count of Unavailable Seconds (UAS) during this interval.
Retransmitted codewords	Retransmitted Codewords.

Table 117: VDSL2 statistics (Continued)

Statistic	Description
Corrected Retransmitted codewords	Retransmitted corrected Codewords.
UnCorrectable Retransmitted codewords	Retransmitted uncorrectable Codewords.
Loss of Power (dying gasps)	Count of Loss of Power (LPR) Seconds during this interval.
phyR Statistics:	
Vtuc PhyRActive	Is this feature active.
Vtuc Retransmitted codewords	ATUC Retransmitted Codewords.
Vtuc Corrected Retransmitted codewords	ATUC Retransmitted corrected Codewords.
Vtuc UnCorrectableRetransmitted codewords	ATUC Retransmitted uncorrectable Codewords.
Vtur Retransmitted codewords	ATUR Retransmitted Codewords.
Vtur Corrected Retransmitted codewords	ATUR Retransmitted corrected Codewords.
Vtur UnCorrectable Retransmitted codewords	ATUR Retransmitted uncorrectable Codewords.
G.INP Statistics:	
Vtuc ginpActive	G.INP/ITU-G.998.4 feature active.
Vtuc Error Free Throughput Rate (LEFTR) Secs	This counter counts the number of seconds that experienced a Low Error Free Throughput Rate (LEFTR), i.e., seconds during which the Error Free Throughput dropped below the configured threshold.
Vtuc Error Free Bits	This counter counts the number of bits that crossed the alpha1/beta1 interface (bits available to carry user payload).
Vtuc Minimum Error Free Throughput Rate	This performance monitoring parameter records the lowest value of Error Free Throughput during the current interval.
Vtur ginpActive	G.INP/ITU-G.998.4 feature active.
Vtur Error Free Throughput Rate (LEFTR) Secs	This counter counts the number of seconds that experienced a Low Error Free Throughput Rate (LEFTR), i.e., seconds during which the Error Free Throughput dropped below the configured threshold.
Vtur Error Free Bits	This counter counts the number of bits that crossed the alpha1/beta1 interface (bits available to carry user payload).
Vtur Minimum Error Free Throughput Rate	This performance monitoring parameter records the lowest value of Error Free Throughput during the current interval.
XTUC PHY Stats:	
serialNumber	The vendor specific string that identifies the vendor equipment.

Table 117: VDSL2 statistics (Continued)

Statistic	Description
vendorID	The vendor ID code is a copy of the binary vendor identification field expressed as readable characters in hexadecimal notation.
versionNumber	The vendor specific version number sent by this Vtu as part of the initialization messages. It is a copy of the binary version number field expressed as readable characters in hexadecimal notation.
curSnrMargin (tenths dB)	Noise Margin as seen by this Vtu with respect to its received signal in 0.25dB. The effective range is -31.75 to +31.75 dB.
currAtn (tenths dB)	Measured difference in the total power transmitted by the peer Vtu and the total power received by this Vtu. The effective range is 0 to +63.75 dB.
currStatus	Indicates current state of the Vtu line. This is a bit-map of possible conditions. The various bit positions are: noDefect There are no defects on the line. lossOfFraming Vtu failure due to not receiving Frame. lossOfSignal Vtu failure due to not receiving Signal. lossOfPower Vtu failure due to loss of power. lossOfSignalQuality Loss of Signal Quality is declared when the Noise Margin falls below the Minimum Noise Margin, or the bit-error-rate exceeds 10^{-7} . lossOfLink Vtu failure due to inability to link with peer Vtu. Set whenever the transceiver is in the 'Warm Start' state. dataInitFailure Vtu failure during initialization due to bit errors corrupting. configInitFailure Vtu failure during initialization due to peer Vtu not able to support requested configuration. protocolInitFailure Vtu failure during initialization due to incompatible protocol used by the peer Vtu. noPeerVtuPresent Vtu failure during initialization due to no activation sequence detected from peer Vtu.
currOutputPwr (tenths dB)	Measured total output power transmitted by this VTU. This is the measurement that was reported during the last activation sequence.
currAttainableRate (bitsPerSec)	Indicates the maximum currently attainable data rate in steps of 1000 bits/second by the Vtu. This value will be equal to or greater than vdslPhysCurrLineRate. Note that for SCM, the minimum and maximum data rates are equal. Note: 1 kbps = 1000 bps.
currLineRate (bitsPerSec)	Indicates the current data rate in steps of 1000 bits/second by the Vtu. This value will be less than or equal to vdslPhysCurrAttainableRate. Note: 1 kbps = 1000 bps

Table 117: VDSL2 statistics (Continued)

Statistic	Description
XTUC CHAN Stats:	
interleaveDelay (tenths milliseconds)	Interleave Delay for this channel. Interleave delay applies only to the interleave (slow) channel and defines the mapping (relative spacing) between subsequent input bytes at the interleave input and their placement in the bit stream at the interleave output. Larger numbers provide greater separation between consecutive input bytes in the output bit stream allowing for improved impulse noise immunity at the expense of payload latency. In the case where the ifType is fast(125), return a value of zero.
crcBlockLength (bytes)	Indicates the length of the channel data-block on which the CRC operates.
currTxRate (bitsPerSec)	Actual transmit data rate on this channel. Note: 1 kbps = 1000 bps.
currTxSlowBurstProt	Actual level of impulse noise (burst) protection for an interleaved (slow) channel. This parameter is not applicable to fast channels. For fast channels, a value of zero shall be returned.
currTxFastFec	Actual Forward Error Correction (FEC) redundancy related overhead for a fast channel. This parameter is not applicable to an interleaved (slow) channel. For interleaved channels, a value of zero shall be returned.
XTUR PHY Stats:	
serialNumber	The vendor specific string that identifies the vendor equipment.
vendorId	The vendor ID code is a copy of the binary vendor identification field expressed as readable characters in hexadecimal notation.
versionNumber	The vendor specific version number sent by this Vtu as part of the initialization messages. It is a copy of the binary version number field expressed as readable characters in hexadecimal notation.
curSnrMargin (tenths dB)	Noise Margin as seen by this Vtu with respect to its received signal in 0.25dB. The effective range is -31.75 to +31.75 dB.
currAtn (tenths dB)	Measured difference in the total power transmitted by the peer Vtu and the total power received by this Vtu. The effective range is 0 to +63.75 dB.

Table 117: VDSL2 statistics (Continued)

Statistic	Description
currStatus.	Indicates current state of the Vtu line. This is a bit-map of possible conditions. The various bit positions are: noDefect There are no defects on the line. lossOfFraming Vtu failure due to not receiving. lossOfSignal Vtu failure due to not receiving. lossOfPower Vtu failure due to loss of power. lossOfSignalQuality Loss of Signal Quality is declared when the Noise Margin falls below the Minimum Noise Margin, or the bit-error-rate exceeds 10^{-7}. lossOfLink Vtu failure due to inability to link with peer Vtu. Set whenever the transceiver is in the 'Warm Start' state. dataInitFailure Vtu failure during initialization due to bit errors corrupting. configInitFailure Vtu failure during initialization due to peer Vtu not able to support requested configuration. protocolInitFailure Vtu failure during initialization due to incompatible protocol used by the peer Vtu. noPeerVtuPresent Vtu failure during initialization due to no activation sequence detected from peer Vtu.
currOutputPwr (tenths dB)	Measured total output power transmitted by this VTU. This is the measurement that was reported during the last activation sequence.
currAttainableRate (bitsPerSec)	Indicates the maximum currently attainable data rate in steps of 1000 bits/second by the Vtu. This value will be equal to or greater than vdslPhysCurrLineRate. Note that for SCM, the minimum and maximum data rates are equal. Note: 1 kbps = 1000 bps
currLineRate (bitsPerSec)	Indicates the current data rate in steps of 1000 bits/second by the Vtu. This value will be less than or equal to vdslPhysCurrAttainableRate. Note: 1 kbps = 1000 bps
XTUR CHAN Stats:	
interleaveDelay (tenths milliseconds)	Interleave Delay for this channel. Interleave delay applies only to the interleave (slow) channel and defines the mapping (relative spacing) between subsequent input bytes at the interleave input and their placement in the bit stream at the interleave output. Larger numbers provide greater separation between consecutive input bytes in the output bit stream allowing for improved impulse noise immunity at the expense of payload latency. In the case where the ifType is fast(125), return a value of zero.
crcBlockLength (bytes)	Indicates the length of the channel data-block on which the CRC operates.

Table 117: VDSL2 statistics (Continued)

Statistic	Description
currTxRate (bitsPerSec)	Actual transmit data rate on this channel. Note: 1 kbps = 1000 bps.
currTxSlowBurstProt	Actual level of impulse noise (burst) protection for an interleaved (slow) channel. This parameter is not applicable to fast channels. For fast channels, a value of zero shall be returned.
currTxFastFec	Actual Forward Error Correction (FEC) redundancy related overhead for a fast channel. This parameter is not applicable to an interleaved (slow) channel. For interleaved channels, a value of zero shall be returned.

ADSL2+ fallback for VDSL2

This section describes ADSL2+ fallback for VDSL2 on the MXK:

- [ADSL2+ fallback for VDSL2 overview, page 1065](#)
- [Case 1: single-service on untagged downlink bridge configurations, page 1066](#)
- [Case 2: single-service on tagged downlink bridge configurations, page 1067](#)
- [Case 3: non-default vpi/vci single-service bridge on tagged or untagged downlink, page 1068](#)
- [Case 4: multi-services on tagged downlink bridge configurations, page 1072](#)
- [Case 5: multi-services on tagged and untagged bridges with non-default vpi/vci, page 1074](#)
- [Case 6: multi-services on tagged bridges for ADSL PTM and VDSL PTM, page 1077](#)

ADSL2+ fallback for VDSL2 overview

VDSL2 on the MXK supports ADSL2+ fallback allowing a VDSL2 loop connection to either a VDSL2 or an ADSL2+ modem.

When the modem supports ATM connections, the default PVC is 0/35. The default PVC can be changed in the **vdsl-config** profile. ATM mode supports only *untagged* frames for both single and multiple service connections. Traffic is segregated by PVCs.

When the modem supports PTM connections, both *tagged* and *untagged* frames for multiple service applications are supported.

The guidelines for ADSL2+ fallback for VDSL2 on the MXK are:

- Single service configurations do not require a PVC in the **bridge add** command. When in ATM mode, the system uses the default PVC of 0/35.
To change the default PVC, configure the **fallbackDefaultVpi** and the **fallbackDefaultVci** parameters in the **vdsl-config** profile to new PVC values.
- Multiple service configurations that define the PVC with the **bridge add** command and the keyword **vc**, rather than in the **vdsl-config** profile, assumes all ATM traffic is *untagged* and sends traffic to a specific bridge based only on the received PVC.
- The system is able to accept tagged ADSL traffic for single service configurations only. The **bridge add** command cannot have the **vc** keyword present. Untagged traffic is also allowed for single PVC ADSL.

- Multiple PVC configurations in PTM mode can have at most one PVC *untagged*. There can not be more than one *untagged* PVC on a single VDSL line.
- When the PVC is specified with the **bridge add** command and an SLAN ID, the bridge must have the *tagged* keyword.



Note: Failure to be consistent with bridging configurations may result in lost services.

The system will not prompt you when you enter inconsistent bridging configurations.

Case 1: single-service on untagged downlink bridge configurations

Creating a single-service untagged downlink bridge configuration

- 1 Create a *uplink* bridge with VLAN ID 200 *tagged*.

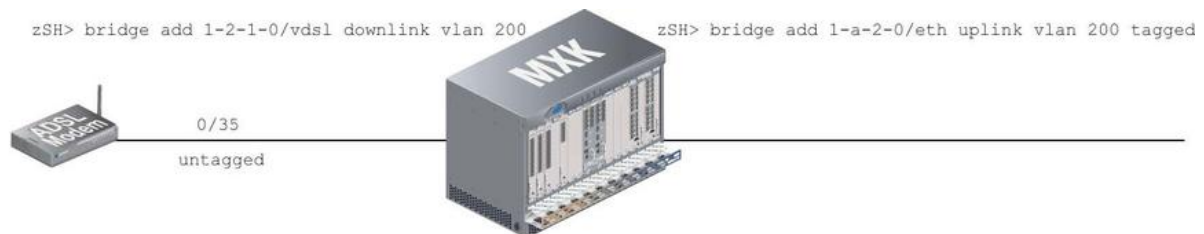
```
zSH> bridge add 1-a-2-0/eth uplink vlan 200 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-200/bridge
Bridge-path added successfully
```

- 2 Create a *untagged* downlink VDSL2 bridge for single-service configurations.

```
zSH> bridge add 1-2-1-0/vdsl downlink vlan 200
Adding bridge on 1-2-1-0/vdsl
Created bridge-interface-record 1-2-1-0-vdsl/bridge
```

- The ADSL modem trains in ATM mode and accepts ATM traffic on 0/35, *untagged*.

Figure 128: MXK to ADSL modem on untagged downlink for single-service



- The VDSL modem trains in PTM mode and accepts PTM packets, *untagged*. In the case of this single-service configuration, the VDSL modem is *not* expecting a VLAN ID.

Figure 129: MXK to VDSL modem on untagged downlink for single-service**3 Verify the bridges.**

```
zSH> bridge show
      Orig
Type VLAN/SLAN  VLAN/SLAN  Physical          Bridge
St  Table Data
-----
dwn                200      1/2/1/0/vdsl      1-2-1-0-vdsl/bridge
DWN
upl                Tagged 200    1/a/2/0/eth       ethernet2-200/bridge
DWN S VLAN 200 default
2 Bridge Interfaces displayed
```

Case 2: single-service on tagged downlink bridge configurations**Creating a single-service tagged downlink bridge configuration****1 Create a uplink bridge with VLAN ID 200 tagged.**

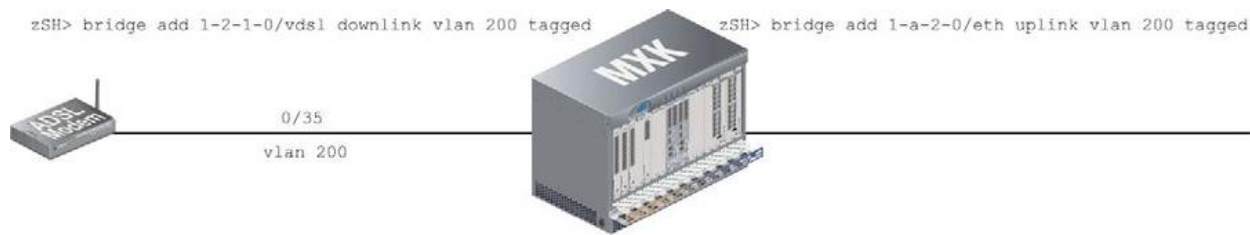
```
zSH> bridge add 1-a-2-0/eth uplink vlan 200 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-200/bridge
Bridge-path added successfully
```

2 Create a tagged downlink VDSL2 bridge for single-service configurations.

```
zSH> bridge add 1-2-1-0/vdsl downlink vlan 200 tagged
Adding bridge on 1-2-1-0/vdsl
Created bridge-interface-record 1-2-1-0-vdsl-200/bridge
```

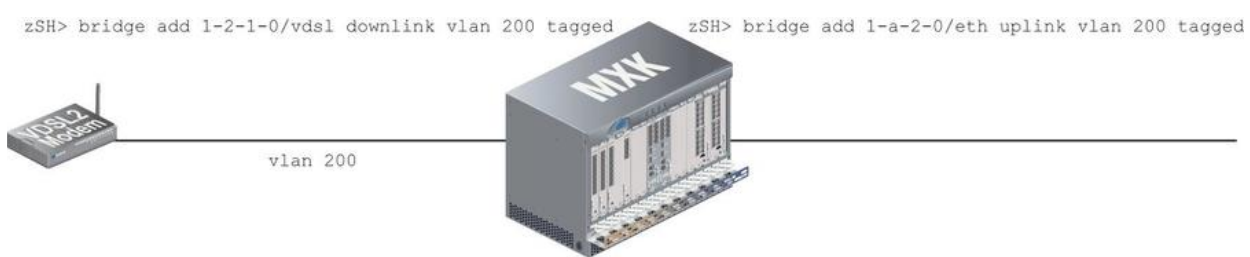
- The ADSL modem trains in ATM mode, accepts ATM traffic on 0/35, and frames will be tagged with the VLAN ID.

Figure 130: MXK to ADSL modem on tagged downlink for single-service



- The VDSL modem trains in PTM mode and traffic is *tagged* with VLAN 200.

Figure 131: MXK to VDSL modem on tagged downlink for single-service



3 Verify the bridges.

```
zSH> bridge show
      Orig
Type VLAN/SLAN  VLAN/SLAN  Physical      Bridge
St  Table Data
-----
dwn          Tagged  200    1/2/1/0/vdsl  1-2-1-0-vdsl-200/bridge
DWN
upl          Tagged  200    1/a/2/0/eth   ethernet2-200/bridge
DWN S VLAN 200 default
2 Bridge Interfaces displayed
```

Case 3: non-default vpi/vci single-service bridge on tagged or untagged downlink

When a single-service bridge is needed and the modem is configured with a *vpi/vci* that is different from the MXK default of 0/35, the *vpi/vci* must be changed in the **vdsl-config** profile, and *not* configured with the **bridge add** command.

Configuring an untagged VDSL2 bridge for single-service with non-default vpi/vci

- 1 Change the default *vpi/vci* in the **vdsl-config** profile.

```
zSH> update vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
```



```

Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}:
line-type: -----> {fastonly}:
vdsl2-profile: -----> {g993-2-17a}:
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
fallbackDefaultVpi: -----> {0}: 0
fallbackDefaultVci: -----> {35}: 36
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Verify the change.

```

zSH> get vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
transmit-mode: -----> {autonegotiatemode}
line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-17a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {36}

```

2 Create an uplink bridge with VLAN ID.

```

zSH> bridge add 1-a-2-0/eth uplink vlan 200 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-200/bridge
Bridge-path added successfully

```

3 Create a *untagged* downlink VDSL2 bridge for single-service configurations with VLAN ID.

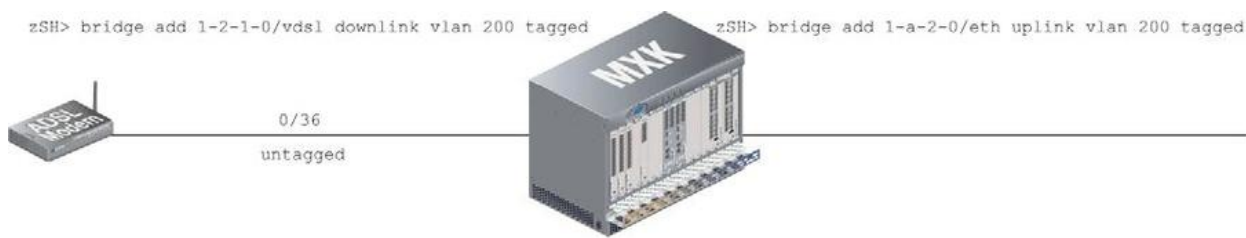
```

zSH> bridge add 1-2-1-0/vdsl downlink vlan 200
Adding bridge on 1-2-1-0/vdsl
Created bridge-interface-record 1-2-1-0-vdsl/bridge

```

- The ADSL modem trains in ATM mode, accepts ATM traffic on 0/36 *untagged*.

Figure 132: MXK to ADSL modem with configured vpi/vc on untagged downlink



- The VDSL modem trains in PTM mode and accepts PTM packets untagged. In the case of this single-service configuration, the VDSL modem is not expecting a VLAN ID.

Figure 133: MXK to VDSL modem on untagged downlink



4 Verify the bridges.

```
zSH> bridge show
      Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge
St  Table Data
-----
dwn          200      1/2/1/0/vdsl  1-2-1-0-vdsl/bridge
DWN
upl          Tagged 200  1/a/2/0/eth   ethernet2-200/bridge
DWN S VLAN 200 default
2 Bridge Interfaces displayed
```

Configuring a tagged VDSL2 bridge for single-service with non-default vpi/vci

1 Change the default vpi/vci in the vdsl-config profile.

```
zSH> update vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}:
line-type: -----> {fastonly}:
vdsl2-profile: -----> {g993-2-17a}:
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
```

```

fallbackDefaultVpi: -----> {0}: 0
fallbackDefaultVci: -----> {35}: 36
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Verify the change.

```

zSH> get vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
transmit-mode: -----> {autonegotiatemode}
line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-17a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {36}

```

2 Create an uplink bridge with VLAN ID.

```

zSH> bridge add 1-a-2-0/eth uplink vlan 200 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-200/bridge
Bridge-path added successfully

```

3 Create a *tagged* downlink VDSL2 bridge for single-service configurations with VLAN ID.

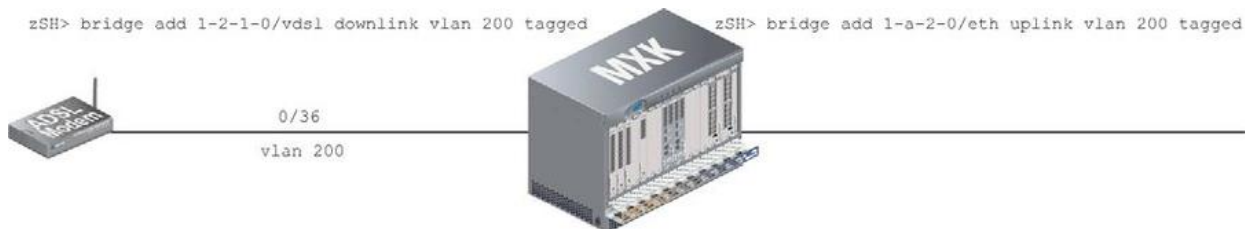
```

zSH> bridge add 1-2-1-0/vdsl downlink vlan 200 tagged
Adding bridge on 1-2-1-0/vdsl
Created bridge-interface-record 1-2-1-0-vdsl-200/bridge

```

- The ADSL modem trains in ATM mode, accepts ATM traffic on 0/36, and frames will be tagged with the VLAN ID 200.

Figure 134: MXK to ADSL modem with configured vpi/vci on tagged downlink



- The VDSL modem trains in PTM mode and accepts PTM packets and traffic is tagged with VLAN ID 200.

Figure 135: MXK to VDSL modem on tagged downlink



4 Verify the bridges.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN  Physical  Bridge  St  Table Data
-----
dwn      Tagged  200  1/2/1/0/vdsl  1-2-1-0-vdsl-200/bridge  DWN
upl      Tagged  200  1/a/2/0/eth   ethernet2-200/bridge     DWN S VLAN 200 default
2 Bridge Interfaces displayed
```

Case 4: multi-services on tagged downlink bridge configurations

Note: For multiple services on the MXK to ADSL modems only, and not VDSL modems, there is special behavior in that although the bridge was configured as *tagged*, the bridge behaves as *untagged*. Traffic is sent to the modem on *vpi/vci*.

Configuring multi-services on tagged downlink bridges

1 Create the uplink bridges with VLAN ID.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 200 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-200/bridge
Bridge-path added successfully

zSH> bridge add 1-a-3-0/eth uplink vlan 300 tagged
Adding bridge on 1-a-3-0/eth
Created bridge-interface-record ethernet3-300/bridge
Bridge-path added successfully
```

2 Create *tagged* downlink bridges with both multiple VCs *and* multiple VLAN IDs.

```
zSH> bridge add 1-2-3-0/vdsl vc 0/35 downlink vlan 200 tagged
Adding bridge on 1-2-3-0/vdsl
Created bridge-interface-record 1-2-3-0-vdsl-0-35-200/bridge

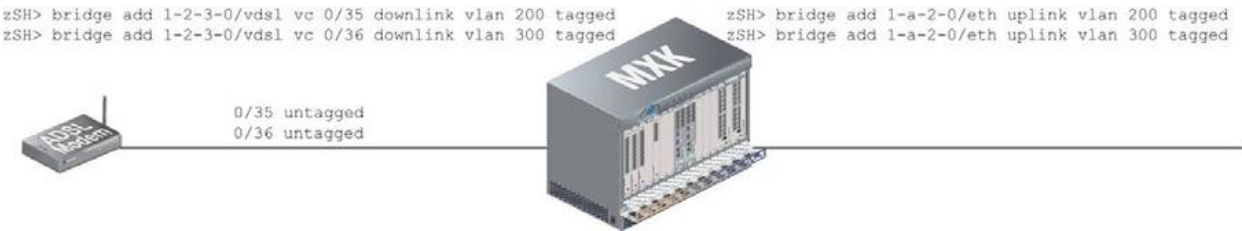
zSH> bridge add 1-2-3-0/vdsl vc 0/36 downlink vlan 300 tagged
Adding bridge on 1-2-3-0/vdsl
Created bridge-interface-record 1-2-3-0-vdsl-0-36-300/bridge
```

For multi-service configurations, the downlink bridge must be *tagged*, because traffic is separated by VLAN IDs for VDSL modems and ATM traffic is separated by the *vpi/vci* for ADSL modems.

When the MXK line links with an ADSL modem and the line is configured for multiple services, the MXK knows not to send tagged traffic to the modem on VLANs but sends *untagged* traffic to the modem using the *vpi/vci* to separate services.

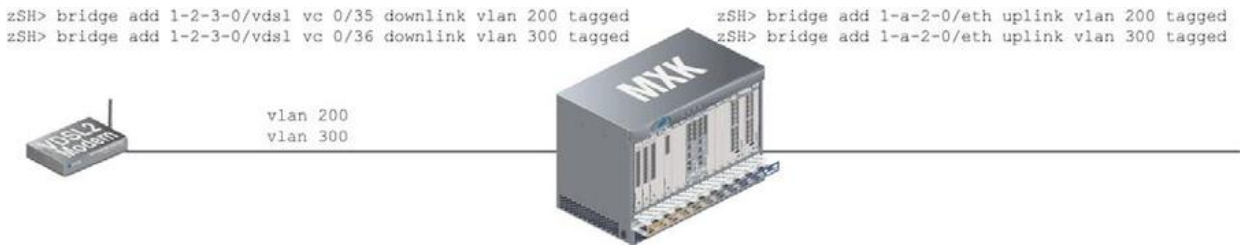
- The ADSL modem accepts ATM traffic on 0/35 and 0/36 *untagged*.

Figure 136: MXK to ADSL modem on tagged downlink for multi-service



- The VDSL modem, accepts PTM packets on VLAN 200 or VLAN 300 *tagged*.

Figure 137: MXK to VDSL modem on tagged downlink for multi-service



3 Verify the bridges.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge
St   Table Data
-----
dwn           Tagged  200    1/2/3/0/vdsl      1-2-3-0-vdsl-0-35-200/
bridge  DWN
dwn           Tagged  300    1/2/3/0/vdsl      1-2-3-0-vdsl-0-36-300/
bridge  DWN
upl           Tagged  200    1/a/2/0/eth       ethernet2-200/bridge
DWN S VLAN 200 default
upl           Tagged  300    1/a/3/0/eth       ethernet3-300/bridge
DWN S VLAN 300 default
4 Bridge Interfaces displayed
```

Case 5: multi-services on tagged and untagged bridges with non-default vpi/vci

This case describes co-existing *tagged* and *untagged* downlink bridges with non-default vpi/vci for multi-services. Multiple service configurations in PTM mode can have at most one service *untagged*. There can not be more than one *untagged* service on a single VDSL line.

Creating multi-services on tagged and untagged bridges with non-default vpi/vci

- 1 Create *tagged* uplink bridges for each VLAN ID in the multi-service configuration.

```
zSH> bridge add 1-a-3-0/eth uplink vlan 200 tagged
Adding bridge on 1-a-3-0/eth
Created bridge-interface-record ethernet3-200/bridge
Bridge-path added successfully
```

```
zSH> bridge add 1-a-4-0/eth uplink vlan 300 tagged
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-300/bridge
Bridge-path added successfully
```

```
zSH> bridge add 1-a-5-0/eth uplink vlan 400 tagged
Adding bridge on 1-a-5-0/eth
Created bridge-interface-record ethernet5-400/bridge
Bridge-path added successfully
```

- 2 Change the default vpi/vci in the **vdsl-config** profile.

```
zSH> update vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}:
line-type: -----> {fastonly}:
vdsl2-profile: -----> {g993-2-17a}:
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
fallbackDefaultVpi: -----> {0}: 0
fallbackDefaultVci: -----> {35}: 36
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Verify the change.

```
zSH> get vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
transmit-mode: -----> {autonegotiatemode}
```

```

line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-17a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {36}

```

3 Create a *untagged* downlink bridge.

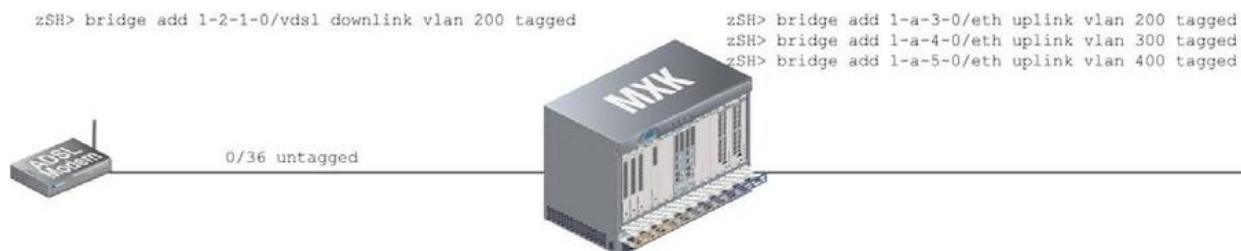
```

zSH> bridge add 1-2-1-0/vdsl downlink vlan 200
Adding bridge on 1-2-1-0/vdsl
Created bridge-interface-record 1-2-1-0-vdsl/bridge

```

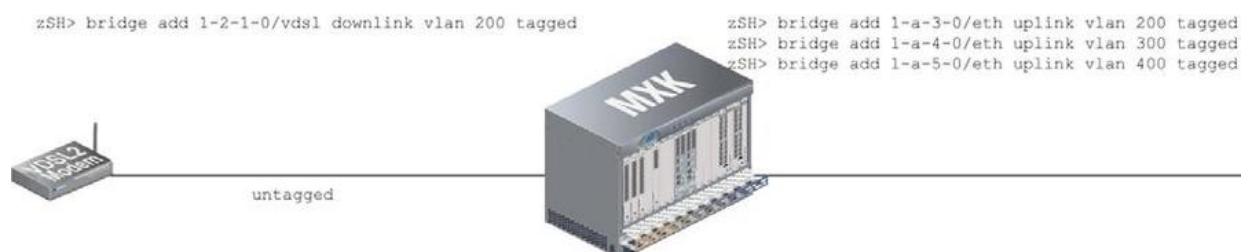
- The ADSL modem trains in ATM mode and accepts ATM traffic on 0/36, *untagged*.

Figure 138: MXK to ADSL modem with configured vpi/vci on untagged downlink



- The VDSL modem trains in PTM mode and accepts PTM packets, *untagged*. In the case of this single-service configuration, the VDSL modem is *not* expecting a VLAN ID.

Figure 139: MXK to VDSL modem on untagged downlink



4 Create the *tagged* downlink bridges with multiple VLAN IDs.

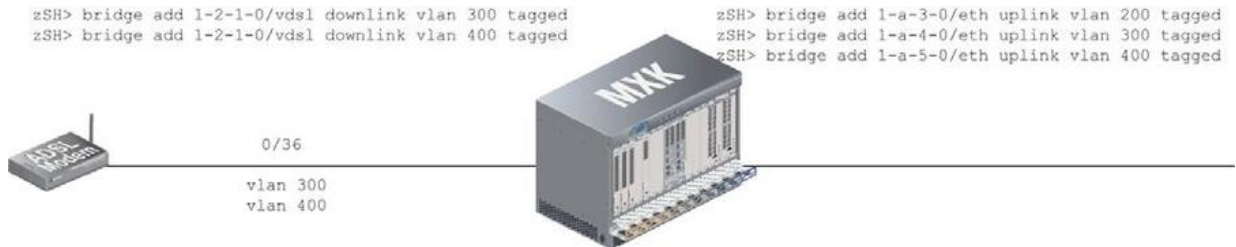
For multi-service configurations, the downlink bridge must be *tagged*, because traffic is separated by VLAN IDs for VDSL modems and ATM traffic is separated by the vpi/vci for ADSL modems. Since a single PVC is used and multi-service traffic is separated by VLAN IDs/tags for both VDSL and ADSL in this scenario, all ATM traffic use the same PVC 0/36.

```
zSH> bridge add 1-2-1-0/vdsl downlink vlan 300 tagged
Adding bridge on 1-2-1-0/vdsl
Created bridge-interface-record 1-2-1-0-vdsl-300/bridge

zSH> bridge add 1-2-1-0/vdsl downlink vlan 400 tagged
Adding bridge on 1-2-1-0/vdsl
Created bridge-interface-record 1-2-1-0-vdsl-400/bridge
```

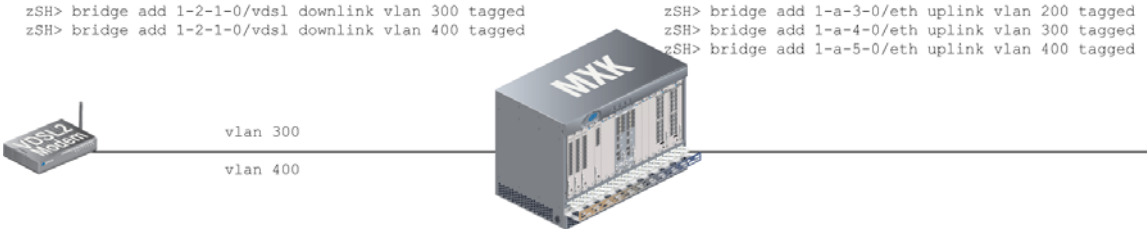
- The ADSL modem trains in ATM mode, accepts ATM traffic on 0/36, and frames will be tagged with the VLAN 300 or VLAN 400.

Figure 140: MXK to ADSL modem with configured vpi/vci on tagged downlinks



- The VDSL modem trains in PTM mode and traffic is *tagged* with VLAN 300 or VLAN 400.

Figure 141: MXK to VDSL modem on tagged downlinks



5 Verify the bridges.

```
zSH> bridge show
```

Orig	Type	VLAN/SLAN	St	VLAN/SLAN	Table	Data	Physical	Bridge
dwn	DWN			200			1/2/1/0/vdsl	1-2-1-0-vdsl/bridge
dwn	DWN			Tagged 300			1/2/1/0/vdsl	1-2-1-0-vdsl-300/bridge
dwn	DWN			Tagged 400			1/2/1/0/vdsl	1-2-1-0-vdsl-400/bridge
upl	DWN S			Tagged 200			1/a/3/0/eth	ethernet3-200/bridge
upl	DWN S			Tagged 300			1/a/4/0/eth	ethernet4-300/bridge


```

upl          Tagged 400    1/a/5/0/eth          ethernet5-400/bridge
UP S VLAN 400 default
6 Bridge Interfaces displayed

```

Case 6: multi-services on tagged bridges for ADSL PTM and VDSL PTM

This case describes *tagged* bridges for multi-services. In this case, both ADSL and VDSL modems accept VLAN IDs and traffic is segregated on VLANs not PVC on the ADSL modem.

Creating multi-services on tagged bridges

- 1 Create *tagged* uplink bridges for each VLAN ID in the multi-service configuration.

```

zSH> bridge add 1-a-2-0/eth uplink vlan 100 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-100/bridge
Bridge-path added successfully

```

```

zSH> bridge add 1-a-3-0/eth uplink vlan 200 tagged
Adding bridge on 1-a-3-0/eth
Created bridge-interface-record ethernet3-200/bridge
Bridge-path added successfully

```

```

zSH> bridge add 1-a-4-0/eth uplink vlan 300 tagged
Adding bridge on 1-a-4-0/eth
Created bridge-interface-record ethernet4-300/bridge
Bridge-path added successfully

```

- 2 Create the *tagged* downlink bridges for each VLAN ID in the multi-service configuration.

```

zSH> bridge add 1-2-1-0/vdsl downlink vlan 100 tagged
Adding bridge on 1-2-1-0/vdsl
Created bridge-interface-record 1-2-1-0-vdsl-100/bridge

```

```

zSH> bridge add 1-2-2-0/vdsl downlink vlan 200 tagged
Adding bridge on 1-2-2-0/vdsl
Created bridge-interface-record 1-2-2-0-vdsl-200/bridge

```

```

zSH> bridge add 1-2-3-0/vdsl downlink vlan 300 tagged
Adding bridge on 1-2-3-0/vdsl
Created bridge-interface-record 1-2-3-0-vdsl-300/bridge

```

Verify the bridges.

```

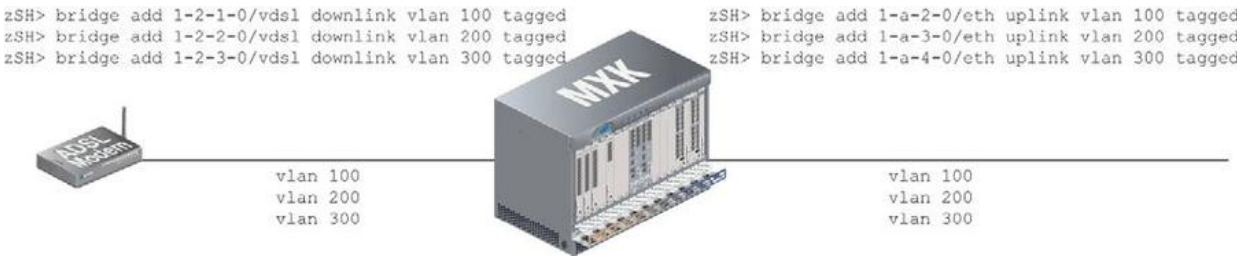
zSH> bridge show
      Orig
Type   VLAN/SLAN  VLAN/SLAN  Physical  Bridge
St  Table Data

```

```
-----
dwn          Tagged 100   1/2/1/0/vdsl      1-2-1-0-vdsl-100/bridge
DWN
dwn          Tagged 200   1/2/2/0/vdsl      1-2-2-0-vdsl-200/bridge
DWN
dwn          Tagged 300   1/2/3/0/vdsl      1-2-3-0-vdsl-300/bridge
DWN
upl          Tagged 100   1/a/2/0/eth        ethernet2-100/bridge
DWN S VLAN 100 default
upl          Tagged 200   1/a/3/0/eth        ethernet3-200/bridge
DWN S VLAN 200 default
upl          Tagged 300   1/a/4/0/eth        ethernet4-300/bridge
DWN S VLAN 300 default
6 Bridge Interfaces displayed
```

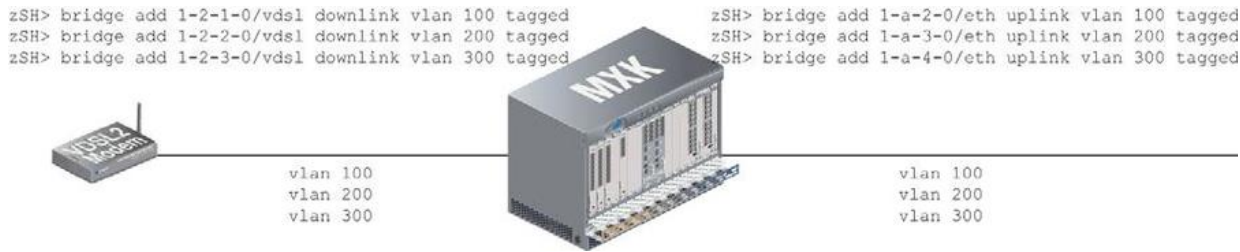
- The ADSL modem trains in PTM mode and traffic is *tagged* with VLAN ID.

Figure 142: MXK to ADSL modem in PTM mode with tagged VLAN IDs



- VDSL modem trains in PTM mode and traffic is *tagged* with VLAN ID.

Figure 143: MXK to VDSL modem in PTM mode with tagged VLAN ID



ADSL2+ and VDSL2 bonding

This section describes ADSL2+ and VDSL2 bonding including:

- [ADSL2+ and VDSL2 bonding rules on 24-port and 48-port VDSL2 cards, page 1079](#)
- [Create gbond groups for VDSL2, page 1082](#)
- [Bridging on ADSL2+ bonding for ADSL, page 1085](#)
- [Bridging on VDSL2 bonding, page 1089](#)

ADSL2+ and VDSL2 bonding rules on 24-port and 48-port VDSL2 cards

This section describes bonding rules for VDSL2 cards:

- [24-port VDSL2 DSP core boundaries and bonding rules, page 1079](#)
- [48-port VDSL2 DSP core boundaries and bonding rules, page 1080](#)
- [Bonding rules common to the 24-port and the 48-port VDSL2 card, page 1081](#)

The 24-port and 48-port VDSL2 card on the MXK supports ADSL2+ or VDSL2 bonding using the `bond add group` and the `bond add member` commands.

Bonding allows two lines to work together as a single line.

24-port VDSL2 DSP core boundaries and bonding rules

The bonding rules for ADSL2+ or VDSL2 24-port VDSL2 card *gbond* groups are:

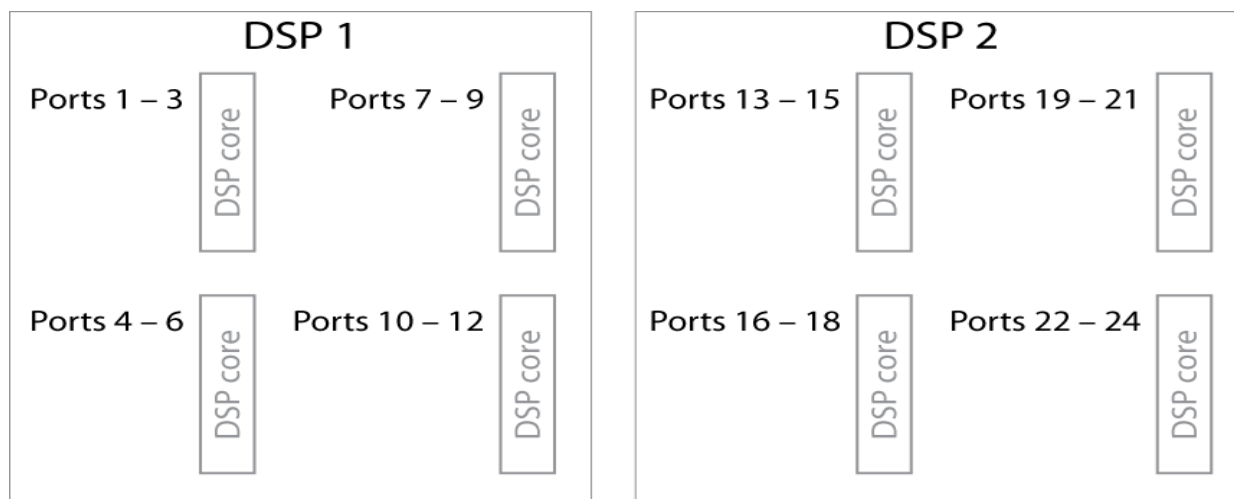
- The valid range for *gbond* groups is 1-24. If an invalid number is used, the CLI returns an error message. For example:

```
zSH> bond add group 1-2-25-0/gbond
Error: group {25} is out-of-range (1..24).
```

- There are two VDSL2 DSP chips per device with ports 1-12 on one chip, and ports 13-24 on the next chip. There are three ports per core and four cores per chip.

Members of *gbond* groups must be consecutive ports which do not cross chip core boundaries. For example, ports 1 and 2 can be in a bond group or ports 2 and 3 but not ports 3 and 4. Ports 4 and 5 or ports 5 and 6 can be bonded, but not ports 6 and 7, and so forth. Refer to [Figure 144](#) for chip core ports.

Figure 144: 24-port VDSL2 DSP core boundaries



48-port VDSL2 DSP core boundaries and bonding rules

The bonding rules specific to ADSL2+ or VDSL2 48-port VDSL2 card *gbond* groups are:

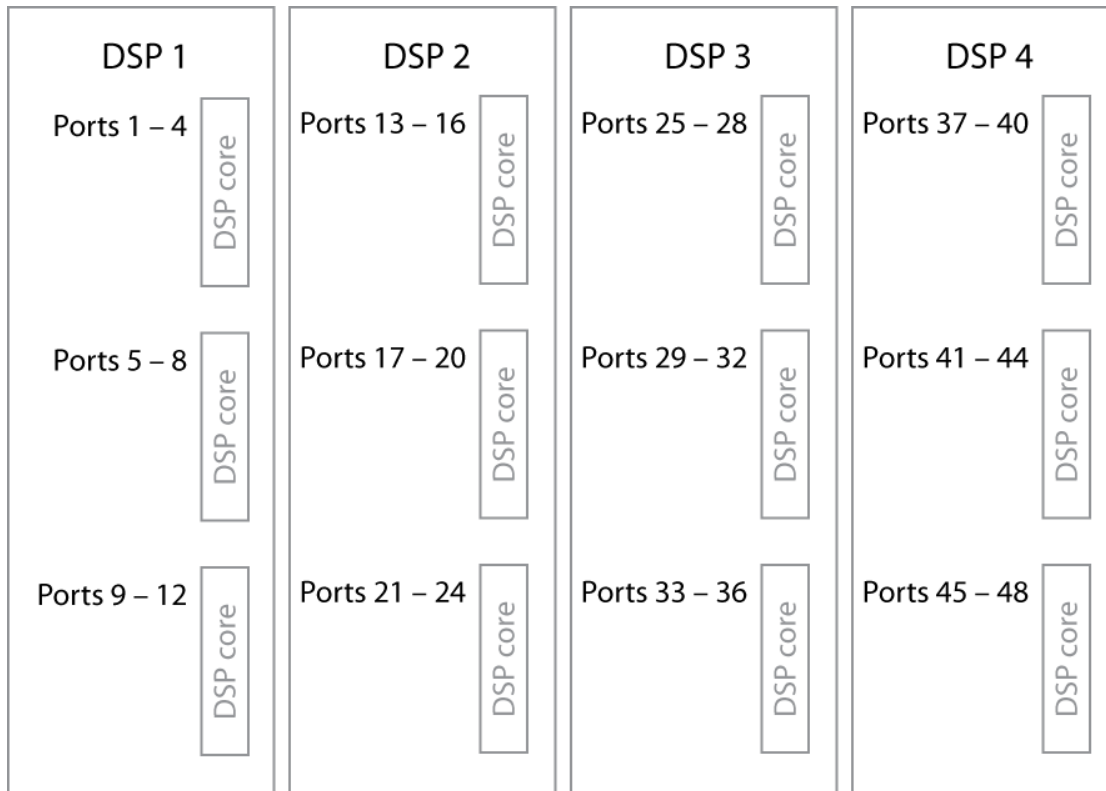
- The valid range for *gbond* groups is 1-48. If an invalid number is used, the CLI returns an error message.

For example:

```
zSH> bond add group 1-12-49-0/gbond
Error: group {49} is out-of-range (1..48).
```

- The 48-port VDSL2 card has four ports per core and bonded ports *must* be in the same core. Refer to [Figure 145](#) for chip core ports.

Figure 145: 48-port VDSL2 DSP core boundaries



- Bonded ports must be consecutive (sequential) and only two ports per bonded groups are allowed with two bond groups allowed per core.

For example, you can bond ports 1 and 2 and 3 and 4 and have two groups. You can also bond ports 2 and 3, however you cannot bond ports 1 and 4 which means this core configuration could only have one bonded group.

Bonding rules common to the 24-port and the 48-port VDSL2 card

The following rules apply to all MXK VDSL2 cards:

- Ports configured with bridges or interfaces will not be allowed to become members of a *gbond* group and the CLI will return an error message.

For example:

```
zSH> bond add member 1-12-1-0/gbond 1-12-1-0/vdsl1
Error: Please remove bridge or IP from the link (1-12-1-0-vdsl1-0-36-998/bridge).
```

- Bridges or interfaces cannot be added/deleted to/from empty *gbond* groups. For example:

```
zSH> bridge add 1-12-1-0/gbond
Cannot add or delete a bridge or interface on a group with no members
(1-12-1-0/gbond)
```

View empty *gbond* group:

```
zSH> bond show group 1-12-1-0/gbond all
Bond Groups
Slot  GrpId  Type    State   Name      Desc
12     1    gbond   OOS     1-12-1-0  -
```

- The last member of a *gbond* group cannot be deleted if a bridge or interface is configured on the *gbond* group. For example,

```
zSH> bond delete member 1-12-1-0/gbond 1-12-1/vdsl1
Cannot delete last group member with a bridge or interface on the group
(1-12-1-0-gbond/bridge)
```

- A *gbond* group cannot be deleted when a bridge or interface is configured on the *gbond* group. For example:

```
zSH> bond delete group 1-12-1-0/gbond
Error: Please remove bridge or IP from the group (1-12-1-0-gbond/bridge).
```

- A bridge or interface cannot be configured on a link that is a member of a *gbond* group. For example:

```
zSH> bridge add 1-12-24-0/vdsl1 downlink vlan 500
Cannot add or delete a bridge or interface on a link in a gbond group (1-2-24-0/
vdsl1)
```

- When ADSL bonded modems are used on VDSL2 ports, the **transmit-mode** parameter in the **vdsl-config** profile must be configured to either *adsl12plusmode*, *adsl12mode*, or *gdmtnode* before the port is added to a *gbond* group.

If this does not occur, the port will not get link. See [Modifying the default vdsl-config parameters on page 1033](#).

- When VDSL2 bonded modems are used on VDSL2 ports, the **transmit-mode** parameter in the **vdsl-config** profile may remain at the default *autonegotiatemode* or set to *vdsl2mode* before the port is added to a *gbond* group.

If this does not occur, the port will not get link. See [Modifying the default vdsl-config parameters on page 1033](#).

The **vdsl2-profile** parameter in the **vdsl-config** profile must also be set to either *8a*, *8b*, *8c*, *8d*, *12b*, and *17a*.



Note: Bonded links on the VDSL2-48-V are capped in all VDSL2 profiles by the Broadcom chipset to downstream rates of 60 Mb and upstream rates of 20 Mb.

Create gbond groups for VDSL2

This section describes *gbond* groups for VDSL2:

- [Bond group creation on 24-port VDSL2 card, page 1083](#)
- [Bond group creation on 48-port VDSL2 card, page 1084](#)



Note: If you are converting from ADSL2+ bonding to VDSL2 bonding (or vice versa) you must delete any bridges/IP/host interfaces, remove the bond group members, change the **transmit-mode** and, if required, the **vdsl2-profile** parameters in the **vdsl-config** profile, re-add the bond group members, and re-provision the bridges/IP/host interfaces.

Bond group creation on 24-port VDSL2 card

Creating a gbond group on a 24-port VDSL2 card

Create a single *gbond* group by first creating the bond group, then adding the members of the *gbond* group.

- 1 Create a *gbond* group with the **bond add group** command and add the members per CLI with the **bond add member** command.

```
zSH> bond add group 1-2-2-0/gbond
```

```
zSH> bond add member 1-2-2-0/gbond 1-2-1-0/vdsl
```

```
zSH> bond add member 1-2-2-0/gbond 1-2-2-0/vdsl
```

- 2 View the *gbond* group and the bond group members with the **bond show group interface/type** command:

```
zSH> bond show group 1-2-2-0/gbond
```

```

Bond Groups
Slot  GrpId  Type      State   Name      Desc
  2      2    gbond     OOS     1-2-2-0   -

Group Members
Slot  Port    Type      State   Name      Desc
  2     1    vdsl     OOS     1-2-1-0   -
  2     2    vdsl     OOS     1-2-2-0   -

```

View all the *gbond* groups that exist on a VDSL2 port with the **bond show slot <slot number>** command.

The *gbond* groups are displayed.

```
zSH> bond show slot 2
```

```

Bond Groups
Slot  GrpId  Type      State   Name      Desc
  2      1    gbond     OOS     1-2-1-0   -
  2      2    gbond     OOS     1-2-2-0   -

```

Deleting a member of a gbond group

Use the **bond delete member** command to delete members of a *gbond* group:

```
zSH> bond delete member 1-2-2-0/gbond 1-2-1-0/vdsl1
zSH> bond delete member 1-2-2-0/gbond 1-2-2-0/vdsl1
Caution: group is now empty!
```

Deleting a gbond group



Note: You cannot delete *gbond* groups that have bridges, IP interfaces, or host interfaces configured on them.

- 1 View the *gbond* group.

```
zSH> bond show group 1-2-2-0/gbond
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
2	2	gbond	OOS	1-2-2-0	-

Group Members					
Slot	Port	Type	State	Name	Desc
2	1	vdsl1	OOS	1-2-1-0	-
2	2	vdsl1	OOS	1-2-2-0	-

- 2 Use the bond delete group command to delete a *gbond* group. The **bond delete group** command deletes *gbond* group and all the members in the *gbond* group.

```
zSH> bond delete group 1-2-2-0/gbond
```

The *gbond group* and all associated members are deleted.

Bond group creation on 48-port VDSL2 card

Creating a gbond group

Create a single *gbond* group by first creating the bond group, then adding the group members of the *gbond* group.

- 1 Create a *gbond* group with the bond add group command and add the members per CLI with the bond add member command.

```
zSH> bond add group 1-12-1-0/gbond
```

```
zSH> bond add member 1-12-1-0/gbond 1-12-25-0/vdsl1
```

```
zSH> bond add member 1-12-1-0/gbond 1-12-26-0/vdsl1
```

- 2 View the *gbond* group and the bond group members with the **bond show group interface/type** command:

```
zSH> bond show group 1-12-1-0/gbond
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
12	1	gbond	OOS	1-12-1-0	-

Group Members					
Slot	Port	Type	State	Name	Desc


```

12          25 vdsl      OOS      1-12-25-0  -
12          26 vdsl      OOS      1-12-26-0  -

```

View all the *gbond* groups that exist on a VDSL2 port with the **bond show slot <slot number>** command.

The *gbond* groups are displayed.

```
zSH> bond show slot 12
```

```

          Bond Groups
Slot  GrpId  Type      State   Name      Desc
12    1      gbond      OOS     1-12-1-0  -

```

Deleting a member of a gbond group

Use the bond delete member command to delete members of a *gbond* group:

```
zSH> bond delete member 1-12-1-0/gbond 1-12-25-0/vdsl
```

```
zSH> bond delete member 1-12-1-0/gbond 1-12-26-0/vdsl
```

Caution: group is now empty!

Deleting a gbond group



Note: You cannot delete *gbond* groups that have bridges, IP interfaces, or host interfaces configured on them.

1 View the *gbond* group.

```
zSH> bond show group 1-12-1-0/gbond
```

```

          Bond Groups
Slot  GrpId  Type      State   Name      Desc
12    1      gbond      OOS     1-12-1-0  -

          Group Members
Slot  Port  Type      State   Name      Desc
12    25 vdsl      OOS     1-12-25-0  -
12    26 vdsl      OOS     1-12-26-0  -

```

2 Use the bond delete group command to delete a *gbond* group. The **bond delete group** command deletes *gbond* group and all the members in the *gbond* group.

```
zSH> bond delete group 1-12-2-0/gbond
```

The *gbond* group and all associated members are deleted.

Bridging on ADSL2+ bonding for ADSL

This section describes bonded bridges including:

- [Bridging on ADSL2+ bonding for ADSL, page 1086](#)

- [Update the vdsl-config file for gbond group members for ADSL2 modems, page 1086](#)
- [Create a tagged downlink bridge on gbond groups with vpi/vci and VLAN ID, page 1088](#)
- [Create a TLS bridge with vpi/vci and VLAN ID, page 1089](#)

Bridging on ADSL2+ bonding for ADSL

This section covers how to:

- [Update the vdsl-config file for gbond group members for ADSL2 modems, page 1095](#)
- [Create a tagged downlink bridge on gbond groups with vpi/vci and VLAN ID, page 1096](#)
- [Create a TLS bridge on gbond groups with vpi/vci and VLAN ID, page 1097](#)



Note: The rules for ADSL2 fallback apply to bonded and non-bonded lines. Depending on the rule, when the default vpi/vci needs to be set in the **vdsl-config** profile, the **vdsl-config** profile for each *gbond* group member must be updated.

See [ADSL2+ fallback for VDSL2, page 1065](#) for usage rules and examples.

Update the vdsl-config file for gbond group members for ADSL2 modems

When ADSL2+ bonded modems are used on VDSL2 ports, the **transmit-mode** parameter in the **vdsl-config** profile must be updated to *adsl2plusmode*, *adsl2mode*, or *gdmtdmode* before adding the port to a *gbond* group.

Updating the vdsl-config file

- 1 Update the **vdsl-config** profile for the first port that will be in the *gbond* group.

```
zSH> update vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}: adsl2plusmode
line-type: -----> {fastonly}:
vdsl2-profile: -----> {g993-2-17a}:
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
```

```

fallbackDefaultVpi: -----> {0}:
fallbackDefaultVci: -----> {35}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Update the next VDSL2 port that will be in the *gbond* group.

```

zSH> update vdsl-config 1-2-2-0/vdsl
vdsl-config 1-2-2-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}: adsl2plusmode
line-type: -----> {fastonly}:
vdsl2-profile: -----> {g993-2-17a}:
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
fallbackDefaultVpi: -----> {0}:
fallbackDefaultVci: -----> {35}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

2 Verify the changes.

```

zSH> get vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
transmit-mode: -----> {adsl2plusmode}
line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-17a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {35}

```

```

zSH> get vdsl-config 1-2-2-0/vdsl
vdsl-config 1-2-2-0/vdsl
transmit-mode: -----> {adsl2plusmode}
line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-17a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {35}

```

Create a tagged downlink bridge on gbond groups with vpi/vci and VLAN ID

You can create a downlink bridge on *gbond* groups when the VDSL2 ports are connected to ADSL bonded capable modems that separate traffic by vpi/vci.



Note: This downlink bridge configuration assumes the *gbond* group is connected to a ADSL bonded modem.

Configuring downlink bridges on a gbond group with vpi/vci and VLAN ID tagged

- 1 Create the *gbond* group. See [Creating a gbond group on a 24-port VDSL2 card on page 1083](#).
- 2 Update the **transmit-mode** parameter in the **vdsl-config** profile to reflect a valid transmit mode for each port in the *gbond* group. See [Update the vdsl-config file for gbond group members for VDSL2 modems on page 1099](#).
- 3 Add the ports with the updated **transmit-mode** parameter to the *gbond* group.
- 4 Create a *tagged* downlink bridge on a *gbond* group and designate the vpi/vci and VLAN ID.

```
zSH> bridge add 1-2-2-0/gbond vc 0/35 downlink vlan 700 tagged
Adding bridge on 1-2-2-0/gbond
Created bridge-interface-record 1-2-2-0-gbond-0-35/bridge
```

- 5 Create an uplink bridge with VLAN ID.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 700 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-700/bridge
Bridge-path added successfully
```

- 6 Verify the bridges.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN      Physical      Bridge
St  Table Data
-----
dwn      Tagged  700      1/2/2/0/gbond      1-2-2-0-gbond-0-35-700/
bridge  UP 00:01:47:31:dc:1a
upl      Tagged  700      1/a/2/0/eth        ethernet2-700/bridge
UP S VLAN 700 default
2 Bridge Interfaces displayed
```

Create a TLS bridge with vpi/vci and VLAN ID



Note: This TLS bridge configuration assumes the *gbond* group is connected to a ADSL bonded modem.

Creating a TLS bridge with vpi/vci and VLAN ID

- 1 Create a subscriber facing TSL bridge with vpi/vci and VLAN ID.

```
zSH> bridge add 1-2-2-0/gbond vc 0/35 tls vlan 1000 tagged
Adding bridge on 1-2-2-0/gbond
Created bridge-interface-record 1-2-2-0-gbond-0-35/bridge
Bridge-path added successfully
```

- 2 Create a network facing TLS bridge with VLAN ID.

```
zSH> bridge add 1-a-2-0/eth tls vlan 1000 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2/bridge
```

- 3 Verify the bridges.

```
zSH> bridge show
      Orig
Type VLAN/SLAN  VLAN/SLAN      Physical      Bridge
St  Table Data
-----
-----
tls          Tagged  1000   1/2/2/0/gbond      1-2-2-0-gbond-0-35-1000/
bridge UP 00:01:47:13:42:27
tls          Tagged  1000   1/a/2/0/eth        ethernet2-1000/bridge
UP 00:01:47:13:42:27
2 Bridge Interfaces displayed
```

Bridging on VDSL2 bonding

This section describes how to:

- [Update the vdsl-config file for gbond group members for VDSL2 modems, page 1099](#)
- [Create a tagged downlink bridge on gbond groups with VLAN ID, page 1102](#)
- [Create a tagged TLS bridge on gbond groups with VLAN ID, page 1102](#)

Update the vdsl-config file for gbond group members for VDSL2 modems

When VDSL2 bonded modems are used on VDSL2 ports, the **transmit-mode** parameter in the **vdsl-config** profile may remain at the default

autonegotiatemode or updated to *vdsl2mode* before the port is added to a *gbond* group.

The **vdsl2-profile** parameter in the **vdsl-config** profile *must* be updated to *g993-2-8a*, *g993-2-8b*, *g993-2-8c*, or *g993-2-8d* for each member of the *gbond* group when connecting to a VDSL2 bonded modem in order to get link.

Updating both the transmit-mode parameter and the vdsl2-profile parameter in the vdsl-config profile

- 1 Update the **transmit-mode** and the **vdsl2-profile** parameters in the **vdsl-config** profile for members of the *gbond* group.

```
zSH> update vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}: vdsl2mode
line-type: -----> {fastonly}:
vdsl2-profile: -----> {g993-2-17a}: g993-2-8a
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
fallbackDefaultVpi: -----> {0}:
fallbackDefaultVci: -----> {35}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

```
zSH> update vdsl-config 1-2-2-0/vdsl
vdsl-config 1-2-2-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}: vdsl2mode
line-type: -----> {fastonly}:
vdsl2-profile: -----> {g993-2-17a}: g993-2-8a
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
fallbackDefaultVpi: -----> {0}:
fallbackDefaultVci: -----> {35}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

- 2 Verify the changes.

```
zSH> get vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
transmit-mode: -----> {vdsl2mode}
line-type: -----> {fastonly}
```

```

vdsl2-profile: -----> {g993-2-8a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {35}

```

```

zSH> get vdsl-config 1-2-2-0/vdsl
vdsl-config 1-2-2-0/vdsl
transmit-mode: -----> {vdsl2mode}
line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-8a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {35}

```

Updating only the vdsl2-profile parameter in the vdsl-config profile

- 1 Update the **vdsl2-profile** parameter in the **vdsl-config** profile for members of the *gbond* group.

```

zSH> update vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}:
line-type: -----> {fastonly}:
vdsl2-profile: -----> {g993-2-17a}: g993-2-8a
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
fallbackDefaultVpi: -----> {0}:
fallbackDefaultVci: -----> {35}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

```

zSH> update vdsl-config 1-2-2-0/vdsl
vdsl-config 1-2-2-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}:
line-type: -----> {fastonly}:
vdsl2-profile: -----> {g993-2-17a}: g993-2-8a
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:

```

```
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
fallbackDefaultVpi: -----> {0}:
fallbackDefaultVci: -----> {35}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

2 Verify the changes.

```
zSH> get vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
transmit-mode: -----> {autonegotiatemode}
line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-8a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {35}
```

```
zSH> get vdsl-config 1-2-2-0/vdsl
vdsl-config 1-2-2-0/vdsl
transmit-mode: -----> {autonegotiatemode}
line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-8a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {35}
```

Create a tagged downlink bridge on gbond groups with VLAN ID



Note: This downlink bridge configuration assumes the *gbond* group is connected to a VDSL2 bonded modem.

You can create a downlink bridge on *gbond* groups when the VDSL2 ports are connected to VDSL2 bonded capable modems.

Configuring tagged downlink bridges on a gbond group with VLAN ID

- 1 Create a tagged downlink bridge on a *gbond* group and designate the VLAN ID.

```
zSH> bridge add 1-2-2-0/gbond downlink vlan 500 tagged
```


Adding bridge on 1-2-2-0/gbond
 Created bridge-interface-record 1-2-2-0-gbond-500/bridge

2 Create an tagged uplink bridge with VLAN ID.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 500 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-500/bridge
Bridge-path added successfully
```

3 Verify the bridges.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN      Physical      Bridge
St   Table Data
-----
dwn           Tagged 500    1/2/2/0/gbond    1-2-2-0-gbond-500/bridge
UP  00:01:47:13:42:27
upl           Tagged 500    1/a/2/0/eth      ethernet2-500/bridge
UP S VLAN 500 default
2 Bridge Interfaces displayed
```

Create a tagged TLS bridge on gbond groups with VLAN ID



Note: This TLS bridge configuration assumes the *gbond* group is connected to a VDSL2 bonded modem.

Creating a TLS bridge with VLAN ID

1 Configure a subscriber facing tagged TLS bridge on a *gbond* group with VLAN ID.

```
zSH> bridge add 1-2-2-0/gbond tls vlan 1500 tagged
Adding bridge on 1-2-2-0/gbond
Created bridge-interface-record 1-2-2-0-gbond/bridge
Bridge-path added successfully
```

2 Configure a networking facing tagged TLS bridge with VLAN ID.

```
zSH> bridge add 1-a-2-0/eth tls vlan 1500 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2/bridge
```

3 Verify the bridges.

```
zSH> bridge show
      Orig
```

Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge
St	Table	Data		

tls		Tagged 1500	1/2/2/0/gbond	1-2-2-0-gbond-1500/bridge
DWN				
tls		Tagged 1500	1/a/2/0/eth	ethernet2-1500/bridge
DWN				
2 Bridge Interfaces displayed				

Bridging on ADSL2+ bonding for ADSL

This section covers:

- [Update the vdsl-config file for gbond group members for ADSL2 modems, page 1095](#)
- [Create a tagged downlink bridge on gbond groups with vpi/vci and VLAN ID, page 1096](#)
- [Create a TLS bridge on gbond groups with vpi/vci and VLAN ID, page 1097](#)



Note: The rules for ADSL2 fallback apply to bonded and non-bonded lines. Depending on the rule, when the default vpi/vci needs to be set in the **vdsl-config** profile, the **vdsl-config** profile for each *gbond* group member must be updated.

See [ADSL2+ fallback for VDSL2, page 1065](#) for usage rules and examples.

Update the vdsl-config file for gbond group members for ADSL2 modems

When ADSL2+ bonded modems are used on VDSL2 ports, the **transmit-mode** parameter in the **vdsl-config** profile must be updated to *adsl2plusmode*, *adsl2mode*, or *gdmtdmode* before adding the port to a *gbond* group.

Updating the vdsl-config file

- 1 Update the **vdsl-config** profile for the first port that will be in the *gbond* group.

```
zSH> update vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}: adsl2plusmode
line-type: -----> {fastonly}:
vdsl2-profile: -----> {g993-2-17a}:
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
fallbackDefaultVpi: -----> {0}:
fallbackDefaultVci: -----> {35}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Update the next VDSL2 port that will be in the *gbond* group.

```

zSH> update vdsl-config 1-2-2-0/vdsl
vdsl-config 1-2-2-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}: adsl2plusmode
line-type: -----> {fastonly}:
vdsl2-profile: -----> {g993-2-17a}:
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
fallbackDefaultVpi: -----> {0}:
fallbackDefaultVci: -----> {35}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

2 Verify the changes.

```

zSH> get vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
transmit-mode: -----> {adsl2plusmode}
line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-17a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {35}

zSH> get vdsl-config 1-2-2-0/vdsl
vdsl-config 1-2-2-0/vdsl
transmit-mode: -----> {adsl2plusmode}
line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-17a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {35}

```

Create a tagged downlink bridge on gbond groups with vpi/vci and VLAN ID

You can create a downlink bridge on *gbond* groups when the VDSL2 ports are connected to ADSL bonded capable modems that separate traffic by vpi/vci.



Note: This downlink bridge configuration assumes the *gbond* group is connected to a ADSL bonded modem.

Configuring downlink bridges on a *gbond* group with vpi/vci and VLAN ID tagged

- 1 Create the *gbond* group. See [Creating a *gbond* group on a 24-port VDSL2 card on page 1083](#).
- 2 Update the **transmit-mode** parameter in the **vdsl-config** profile to reflect a valid transmit mode for each port in the *gbond* group. See [Update the *vdsl-config* file for *gbond* group members for VDSL2 modems on page 1099](#).
- 3 Add the ports with the updated **transmit-mode** parameter to the *gbond* group.
- 4 Create a *tagged* downlink bridge on a *gbond* group and designate the vpi/vci and VLAN ID.

```
zSH> bridge add 1-2-2-0/gbond vc 0/35 downlink vlan 700 tagged
Adding bridge on 1-2-2-0/gbond
Created bridge-interface-record 1-2-2-0-gbond-0-35/bridge
```

- 5 Create an uplink bridge with VLAN ID.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 700 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-700/bridge
Bridge-path added successfully
```

- 6 Verify the bridges.

```
zSH> bridge show
      Orig
Type VLAN/SLAN  VLAN/SLAN      Physical      Bridge
St  Table Data
-----
dwn          Tagged  700    1/2/2/0/gbond    1-2-2-0-gbond-0-35-700/
bridge  UP 00:01:47:31:dc:1a
upl          Tagged  700    1/a/2/0/eth      ethernet2-700/bridge
UP S VLAN 700 default
2 Bridge Interfaces displayed
```

Create a TLS bridge on *gbond* groups with vpi/vci and VLAN ID



Note: This TLS bridge configuration assumes the *gbond* group is connected to a ADSL bonded modem.

Creating a TLS bridge with vpi/vci and VLAN ID

- 1 Create a subscriber facing TSL bridge with vpi/vci and VLAN ID.

```
zSH> bridge add 1-2-2-0/gbond vc 0/35 tls vlan 1000 tagged
Adding bridge on 1-2-2-0/gbond
Created bridge-interface-record 1-2-2-0-gbond-0-35/bridge
Bridge-path added successfully
```

- 2 Create a network facing TLS bridge with VLAN ID.

```
zSH> bridge add 1-a-2-0/eth tls vlan 1000 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2/bridge
```

- 3 Verify the bridges.

```
zSH> bridge show
Orig
Type VLAN/SLAN  VLAN/SLAN      Physical      Bridge
St  Table Data
-----
-----
tls          Tagged  1000   1/2/2/0/gbond      1-2-2-0-gbond-0-35-1000/
bridge UP 00:01:47:13:42:27
tls          Tagged  1000   1/a/2/0/eth        ethernet2-1000/bridge
UP 00:01:47:13:42:27
2 Bridge Interfaces displayed
```

Bridging on VDSL2 bonding for VDSL

This section describes:

- [Update the vdsl-config file for gbond group members for VDSL2 modems, page 1099](#)
- [Create a tagged downlink bridge on gbond groups with VLAN ID, page 1102](#)
- [Create a tagged TLS bridge on gbond groups with VLAN ID, page 1102](#)

Update the vdsl-config file for gbond group members for VDSL2 modems

When VDSL2 bonded modems are used on VDSL2 ports, the **transmit-mode** parameter in the **vdsl-config** profile may remain at the default *autonegotiatemode* or updated to *vdsl2mode* before the port is added to a *gbond* group.

The **vdsl2-profile** parameter in the **vdsl-config** profile *must* be updated to *g993-2-8a*, *g993-2-8b*, *g993-2-8c*, or *g993-2-8d* for each member of the *gbond* group when connecting to a VDSL2 bonded modem in order to get link.

Updating both the transmit-mode parameter and the vdsl2-profile parameter in the vdsl-config profile

- 1 Update the **transmit-mode** and the **vdsl2-profile** parameters in the **vdsl-config** profile for members of the *gbond* group.

```
zSH> update vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}: vdsl2mode
line-type: -----> {fastonly}:
vdsl2-profile: -----> {g993-2-17a}: g993-2-8a
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
fallbackDefaultVpi: -----> {0}:
fallbackDefaultVci: -----> {35}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

```
zSH> update vdsl-config 1-2-2-0/vdsl
vdsl-config 1-2-2-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}: vdsl2mode
line-type: -----> {fastonly}:
```

```

vdsl2-profile: -----> {g993-2-17a}: g993-2-8a
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
fallbackDefaultVpi: -----> {0}:
fallbackDefaultVci: -----> {35}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

2 Verify the changes.

```

zSH> get vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
transmit-mode: -----> {vdsl2mode}
line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-8a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {35}

```

```

zSH> get vdsl-config 1-2-2-0/vdsl
vdsl-config 1-2-2-0/vdsl
transmit-mode: -----> {vdsl2mode}
line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-8a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {35}

```

Updating only the vdsl2-profile parameter in the vdsl-config profile

- 1 Update the **vdsl2-profile** parameter in the **vdsl-config** profile for members of the *gbond* group.

```

zSH> update vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}:
line-type: -----> {fastonly}:
vdsl2-profile: -----> {g993-2-17a}: g993-2-8a
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:

```



```

trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
fallbackDefaultVpi: -----> {0}:
fallbackDefaultVci: -----> {35}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

zSH> update vdsl-config 1-2-2-0/vdsl
vdsl-config 1-2-2-0/vdsl
Please provide the following: [q]uit.
transmit-mode: -----> {autonegotiatemode}:
line-type: -----> {fastonly}:
vdsl2-profile: -----> {g993-2-17a}: g993-2-8a
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu32}:
trellis-enabled: -----> {true}:
rs-enabled: -----> {true}:
psd-shape: -----> {region-a-eu-32}:
fallbackDefaultVpi: -----> {0}:
fallbackDefaultVci: -----> {35}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

2 Verify the changes.

```

zSH> get vdsl-config 1-2-1-0/vdsl
vdsl-config 1-2-1-0/vdsl
transmit-mode: -----> {autonegotiatemode}
line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-8a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {35}

zSH> get vdsl-config 1-2-2-0/vdsl
vdsl-config 1-2-2-0/vdsl
transmit-mode: -----> {autonegotiatemode}
line-type: -----> {fastonly}
vdsl2-profile: -----> {g993-2-8a}
adslAnnexMModeEnabled: -----> {false}
adslAnnexMPsdMask: -----> {eu32}
trellis-enabled: -----> {true}
rs-enabled: -----> {true}
psd-shape: -----> {region-a-eu-32}
fallbackDefaultVpi: -----> {0}
fallbackDefaultVci: -----> {35}

```

Create a tagged downlink bridge on gbond groups with VLAN ID



Note: This downlink bridge configuration assumes the *gbond* group is connected to a VDSL bonded modem.

You can create a downlink bridge on *gbond* groups when the VDSL2 ports are connected to VDSL2 bonded capable modems.

Configuring tagged downlink bridges on a gbond group with VLAN ID

- 1 Create a tagged downlink bridge on a *gbond* group and designate the VLAN ID.

```
zSH> bridge add 1-2-2-0/gbond downlink vlan 500 tagged
Adding bridge on 1-2-2-0/gbond
Created bridge-interface-record 1-2-2-0-gbond-500/bridge
```

- 2 Create an tagged uplink bridge with VLAN ID.

```
zSH> bridge add 1-a-2-0/eth uplink vlan 500 tagged
Adding bridge on 1-a-2-0/eth
Created bridge-interface-record ethernet2-500/bridge
Bridge-path added successfully
```

- 3 Verify the bridges.

```
zSH> bridge show
      Orig
Type  VLAN/SLAN  VLAN/SLAN  Physical  Bridge
St   Table Data
-----
dwn           Tagged  500    1/2/2/0/gbond    1-2-2-0-gbond-500/bridge
UP  00:01:47:13:42:27
upl           Tagged  500    1/a/2/0/eth      ethernet2-500/bridge
UP S VLAN 500 default
2 Bridge Interfaces displayed
```

Create a tagged TLS bridge on gbond groups with VLAN ID



Note: This TLS bridge configuration assumes the *gbond* group is connected to a VDSL bonded modem.

Creating a TLS bridge with VLAN ID

- 1 Configure a subscriber facing tagged TLS bridge on a *gbond* group with VLAN ID.

```
zSH> bridge add 1-2-2-0/gbond tls vlan 1500 tagged
Adding bridge on 1-2-2-0/gbond
```

Created bridge-interface-record 1-2-2-0-gbond/bridge
 Bridge-path added successfully

2 Configure a networking facing tagged TLS bridge with VLAN ID.

zSH> **bridge add 1-a-2-0/eth tls vlan 1500 tagged**
 Adding bridge on 1-a-2-0/eth
 Created bridge-interface-record ethernet2/bridge

3 Verify the bridges.

zSH> **bridge show**

Orig	Type	VLAN/SLAN	VLAN/SLAN	Physical	Bridge
St	Table	Data			

tls		Tagged	1500	1/2/2/0/gbond	1-2-2-0-gbond-1500/bridge
DWN					
tls		Tagged	1500	1/a/2/0/eth	ethernet2-1500/bridge
DWN					

2 Bridge Interfaces displayed

Upstream Power Backoff (UPBO) for VDSL2

When VDSL2 CPE devices connected to an MXK are set to transmit the same power level, then the amount of power received from the CPE device at the MXK VDSL2 card on a short loop will be quite a bit higher than the power level received from a CPE device on a long loop, causing the short loop signal to interfere with the signals on the long loop, resulting in unreliable operation of VDSL2 on the long loop. Upstream Power Backoff (UPBO) ensures that the upstream frequencies received at the MXK VDSL2 card are equalized so that the overall power levels at the VDSL line input in the upstream frequencies are approximately equal, independent of the actual loop lengths.

UPBO is normally used to attenuate upstream power on the short loops in order to equalize the received power levels between short and long loops in the same binder group. This attenuation will achieve a better balance between the data rates available on short and long loops while providing more reliable VDSL2 operation on long loops.

Enabling UPBO has two effects:

- Reduces the amount of signal or power levels in the upstream direction on the loop. This backing off of power has the effect of reducing the overall Signal to Noise Ratio (SNR) on this loop, which in turn will reduce the overall data rates that can be reliably achieved on the loop. A reduction in data rate on short loops is normally not too critical or significant since the SNR values are greatest on short loops.
- Due to the reduced signal / power levels on a loop which has UPBO applied, the Far End Cross Talk (FEXT) on any adjacent loops will be reduced. The reduced FEXT effectively increases the SNR on adjacent loops and allows for increased data rates / reliable operation of VDSL2 on these loops than would otherwise be possible. Therefore, by applying UPBO on a short loop, you can improve reliable operation of any adjacent long loops that exist in the same binder group.

Enabling UPBO

To enable UPBO, you change the **pbo-control** parameter to **auto** and select the **pbo-psd-template** per configured link.

1 List the **vdsl-cpe-config**

```
zSH> list vdsl-cpe-config
vdsl-cpe-config 1-12-1-0/vdsl
vdsl-cpe-config 1-12-2-0/vdsl
vdsl-cpe-config 1-12-3-0/vdsl
vdsl-cpe-config 1-12-4-0/vdsl
.....
vdsl-cpe-config 1-12-21-0/vdsl
vdsl-cpe-config 1-12-22-0/vdsl
vdsl-cpe-config 1-12-23-0/vdsl
vdsl-cpe-config 1-12-24-0/vdsl
24 entries found.
```

- 2 Update the **pbo-control** parameter and, if desired, change the **pbo-psd-template** value.

```

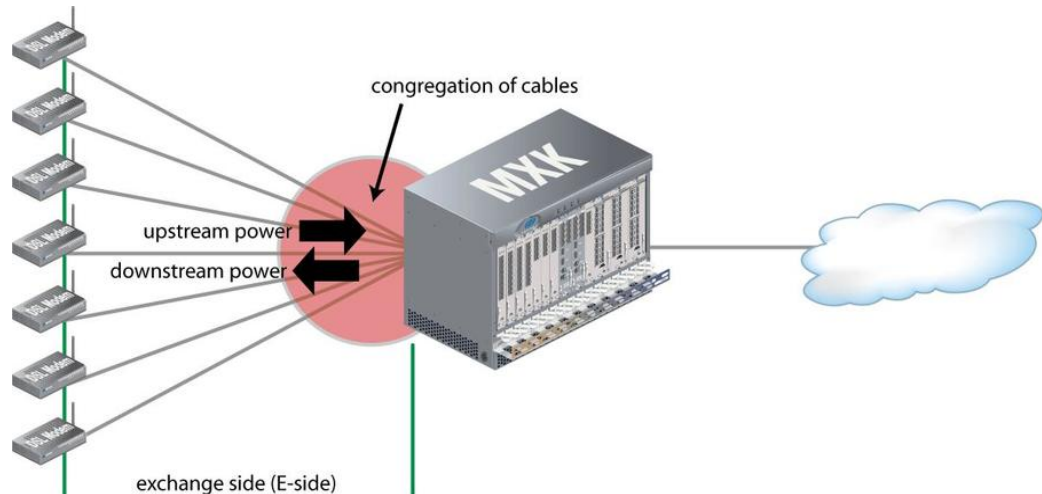
zSH> update vdsl-cpe-config 1-12-1-0/vdsl
vdsl-cpe-config 1-12-1-0/vdsl
Please provide the following: [q]uit.
fastMaxTxRate: -----> {200000}:
fastMinTxRate: -----> {0}:
interleaveMaxTxRate: -----> {200000}:
interleaveMinTxRate: -----> {0}:
rateMode: -----> {dynamic}:
maxPower: -----> {130}:
maxSnrMgn: -----> {127}:
minSnrMgn: -----> {0}:
targetSnrMgn: -----> {60}:
pbo-control: -----> {disabled}: auto
pbo-psd-template: -----> {ansi-f}: ansi-a
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minDownshiftTime: -----> {30}:
minUpshiftTime: -----> {30}:
bitSwap: -----> {enabled}:
minINP: -----> {twosymbols}:
maxInterleaveDelay: -----> {20}:
phyRSupport: -----> {enable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Downstream Power Backoff (DPBO)

The MXK supports shaped downstream power backoff (DPBO) as described in ITU-T G.997. Like upstream power backoff, the idea of DPBO is to limit the interference generated where the cable congregates at the central office or street cabinet while still providing enough power to properly receive data from the far end device.

Figure 146: Both upstream and downstream power backoff reduce the power and hence the interference where the cables congregate at the CO or cabinet



DPBO is supported on the ADSL2+ and VDSL2 line cards.

DPBO configuration data should be calculated for each network using formula provided in ITU-T G.997.1 or those provided in a country specific specification. Typically the model is consistent across a given network with just the electrical length value ESEL changing for different nodes.

DPBO configuration settings follow the ITU-T G.997.1 standard. The mechanism for applying DPBO settings to DSL links is via the **dsl-config** profile. The **dsl-config** profile sets an index for a set of profiles to define DPBO:

- **dpbo-profile**

General E-side (exchange side or what can also be called the subscriber side) parameters for DPBO. These parameters configure the basic signal shape which is adjusted through the three sets of masks.

- **dpbo-epsd**

The assumed Power Spectrum Density (PSD) mask. This mask defines the exchange side PSD mask.

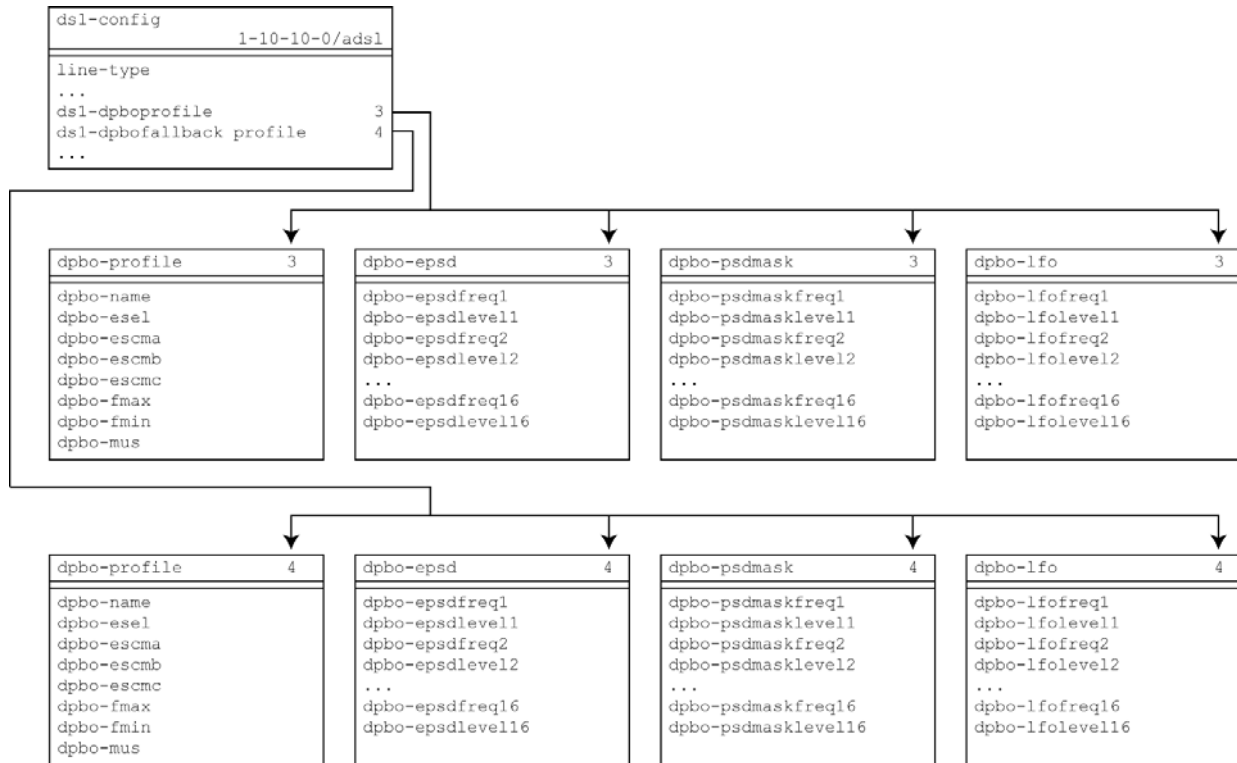
- **dpbo-psdmask**

The maximum PSD mask. This mask defines the maximum PSD levels.

- **dpbo-lfo**

The low frequency override. This set of breakpoints override the PSD mask at low frequencies.

Figure 147: The mechanism for defining DPBO and the associated masks has the same index for the dpbo-profile, dpbo-epsd, dpbo-psdmask and dpbo-lfo profiles



The **dpbo-escma**, **dpbo-escmb**, and **dpbo-escmc** configuration parameters are three scalars which define a cable model that describes the frequency dependent loss of E-side cables using the formula:

$$ESCM(f) = (DPBOESCMA + DPBOESCMB * \sqrt{f} + DPBOESCMC * f) * DPBOESEL$$

where ESCM is expressed in dB and f is expressed in MHz.



Note: DPBO configuration data should be calculated for each network using formula provided in ITU-T G.997.1 or those provided in a country specific specification. Typically the model is consistent across a given network with just the electrical length value ESEL changing for different nodes.

Table 118: dpbo-profile

Parameter	Description
dpbo-name	User configurable name

Table 118: dpbo-profile (Continued)

Parameter	Description
dpbo-esel	E-Side Electrical Length. Defines the assumed electrical length of cables (Exchange side cables) connecting exchange based DSL services to a remote flexibility point (cabinet) that hosts the MXK that is subject to spectrally shaped downstream power back-off depending on this length. For this parameter the electrical length is defined as the loss (in dB) of an equivalent length of hypothetical cable at a reference frequency defined by the network operator or in spectrum management regulations. An unsigned integer representing an electrical length from 0 dB to 255.5 dB in steps of 0.1 dB. All values in the range are valid.
dpbo-escma	E-Side Cable Model Parameter A. An unsigned integer representing a scalar value from -1 to 1.5 in steps of 1/256 dB.
dpbo-escmb	E-Side Cable Model Parameter B. An unsigned integer representing a scalar value from -1 to 1.5 in steps of 1/256 dB.
dpbo-escmc	E-Side Cable Model Parameter C. An unsigned integer representing a scalar value from -1 to 1.5 in steps of 1/256 dB.
dpbo-fmax	Maximum frequency. Defines the maximum frequency at which DPBO may be applied. It ranges from 138 kHz to 29997.75 kHz in steps of 4.3125 kHz.
dpbo-fmin	Minimum frequency. Defines the minimum frequency from which the DPBO shall be applied. It ranges from 0 kHz to 8832 kHz in steps of 4.3125 kHz.
dpbo-mus	Minimum Usable Signal. Defines the assumed Minimum Usable receive PSD mask (in dBm/Hz) for exchange based services, used to modify parameter dpbo-fmax . It represents a PSD mask level from 0 dBm/Hz to -127.5 dBm/Hz in steps of 0.1 dB. All values in the range are valid. NOTE: The PSD mask level is 3.5 dB above the signal PSD level.

The **dpbo-epsd** profile allows configuration of the ITU-T G.997.1, paragraph 7.3.1.2.13 Downstream Power Back-Off DPBO Assumed exchange PSD mask breakpoints (referred to as DPBOEPSD in the ITU-T document). There are a maximum of 16 breakpoints. This set of breakpoints defines the PSD mask that is assumed to be permitted at the exchange. If an EPSD profile is not created or configured for a default DPBO data set, the default PSD shape is assumed.

Table 119: dpbo-epsd profile

Parameter	Description
dpbo-epsdfreq1...16	The dpbo-epsdfreqX configuration parameters define the assumed exchange PSD mask (DPBOEPSD) Breakpoint Frequency for the specified breakpoint. Up to 16 may be defined. For convenience, this value is set in KHz. A zero frequency value disables the breakpoint.

Table 119: dpbo-epsd profile (Continued)

Parameter	Description
dpbo-epsdlevel11...16	The dpbo-epsdlevelX configuration parameters define the assumed exchange PSD mask (DPBOEPSD) Breakpoint PSD Level for the specified breakpoint. Up to 16 may be defined.

The **dpbo-psdmask** profile allows configuration of the ITU-T G.997.1, paragraph 7.3.1.2.13 Downstream Power Back-Off DPBO maximum PSD mask downstream breakpoints (referred to as DPBOPSDMASKds in the ITU-T document). There are a maximum of 16 breakpoints. If a limiting PSD mask profile is not created or configured for a default DPBO data set, for ADSL the assumed default is the set of EPSD breakpoints and for VDSL the default VDSL limiting mask is assumed. Often this set is not configured for VDSL lines.

Table 120: dpbo-psdmask

Parameter	Description
dpbo-psdmaskfreq1...16	The dpbo-psdmaskfreqX configuration parameters define the assumed exchange PSD mask (DPBOEPSD) Breakpoint Frequency for the specified breakpoint. Up to 16 may be defined. For convenience, this value is set in KHz. A zero frequency value disables the breakpoint.
dpbo-psdmasklevel11...16	The dpbo-psdmasklevelX configuration parameters define the assumed exchange PSD mask (DPBOEPSD) Breakpoint PSD Level for the specified breakpoint. Up to 16 may be defined.

The **dpbo-lfo** profile allows configuration of the ITU-T G.997.1, paragraph 7.3.1.2.13 Downstream Power Back-Off DPBO low frequency override breakpoints (referred to as DPBOLFO in the ITU-T document). There are a maximum of 16 breakpoints. This set of breakpoints defines the PSD mask that overrides DPBO at low frequencies. If LFO profile is not created or configured for a default DPBO data set no default is needed or provided.

Table 121: dpbo-lfo

Parameter	Description
dpbo-lfofreq1...16	The dpbo-lfofreqX configuration parameters define the assumed exchange PSD mask (DPBOEPSD) Breakpoint Frequency for the specified breakpoint. Up to 16 may be defined. For convenience, this value is set in KHz. A zero frequency value disables the breakpoint.
dpbo-lfolevel11...16	The dpbo-lfolevelX configuration parameters define the assumed exchange PSD mask (DPBOEPSD) Breakpoint PSD Level for the specified breakpoint. Up to 16 may be defined.

Example calculating E-Side Cable Model parameters

You need to provide the desired ESEL (dpbo-esel) and cable model values (dpbo-escma, dpbo-escmb, dpbo-escmc) as ESEL values vary from node to node and the ESCM values vary from network to network and country to country.

Table 122: Example calculating the E-side Cable Model parameters

Parameter	Desired Value	Formula Example
ESCMA (dpbo-escma)	0.0546875 dB	$0.0546875 - (-1) = 1.0546875$ dB above -1 dB $1.0546875 \times 256 = 270$ steps
ESCMB (dpbo-escmb)	0.9140625 dB	$0.9140625 - (-1) = 1.9140625$ dB above -1 dB $1.9140625 \times 256 = 490$ steps
ESCMC (dpbo-escmc)	0.03125 dB	$0.03125 - (-1) = 1.03125$ dB above -1 dB $1.03125 \times 256 = 264$ steps

The calculations shown in the table above would then be entered in the dpbo-profile as shown below (bolded). These parameters follow the standards from the ITU-T G.997.1 standards.

```

zSH> new dpbo-profile 3
dpbo-profile 3

Please provide the following: [q]uit.
dpbo-name: --> {dpboprof}: VDSL2_DPBO
dpbo-esel: --> {0}: 270
dpbo-escma: -> {0}: 270
dpbo-escmb: -> {0}: 490
dpbo-escmc: -> {0}: 264
dpbo-fmax: --> {32}: 512
dpbo-fmin: --> {0}: 64
dpbo-mus: ---> {0}: -1110
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
zSH> new dpbo-epsd 3

```

Applying DPBO to a VDSL2 interface

Notice that there is a single index which defines the DPBO associated profiles.

- 1 Create a new **dpbo-profile** and configure with the appropriate settings

```

zSH> new dpbo-profile 3

dpbo-profile 3
Please provide the following: [q]uit.

```

```

dpbo-name: --> {dpboprof}: VDSL2_DPBO
dpbo-esel: --> {0}: 270
dpbo-escma: -> {0}: 270
dpbo-escmb: -> {0}: 490
dpbo-escmc: -> {0}: 264
dpbo-fmax: --> {32}: 512
dpbo-fmin: --> {0}: 64
dpbo-mus: ---> {0}: -1110
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```

2 Create a new **dpbo-epsd** profile and configure with the appropriate settings

```
zSH> new dpbo-epsd 3
```

```

dpbo-epsd 3
Please provide the following: [q]uit.
dpbo-epsdfreq1: ---> {0}: 276
dpbo-epsdlevel1: --> {0}: -400
dpbo-epsdfreq2: ---> {0}: 2204
dpbo-epsdlevel2: --> {0}: -400
dpbo-epsdfreq3: ---> {0}:
dpbo-epsdlevel3: --> {0}:
dpbo-epsdfreq4: ---> {0}:
dpbo-epsdlevel4: --> {0}:
dpbo-epsdfreq5: ---> {0}:
dpbo-epsdlevel5: --> {0}:
dpbo-epsdfreq6: ---> {0}:
dpbo-epsdlevel6: --> {0}:
dpbo-epsdfreq7: ---> {0}:
dpbo-epsdlevel7: --> {0}:
dpbo-epsdfreq8: ---> {0}:
dpbo-epsdlevel8: --> {0}:
dpbo-epsdfreq9: ---> {0}:
dpbo-epsdlevel9: --> {0}:
dpbo-epsdfreq10: --> {0}:
dpbo-epsdlevel10: -> {0}:
dpbo-epsdfreq11: --> {0}:
dpbo-epsdlevel11: -> {0}:
dpbo-epsdfreq12: --> {0}:
dpbo-epsdlevel12: -> {0}:
dpbo-epsdfreq13: --> {0}:
dpbo-epsdlevel13: -> {0}:
dpbo-epsdfreq14: --> {0}:
dpbo-epsdlevel14: -> {0}:
dpbo-epsdfreq15: --> {0}:
dpbo-epsdlevel15: -> {0}:
dpbo-epsdfreq16: --> {0}:
dpbo-epsdlevel16: -> {0}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```

- 3 <Optional> Create new **dpbo-psdmask** profile and **dpbo-lfo** profile and configure with the appropriate settings

The **dpbo-psdmask** profile and **dpbo-lfo** profiles are not required. If one or the other is not created a default mask for their properties is not used. The default mask is not needed.

- 4 Apply the DPBO settings to a VDSL2 interface

```
zSH> update dsl-config 1-2-10-0/vdsl1

dsl-config 1-2-10-0/vdsl1
Please provide the following: [q]uit.
line-type: -----> {vdsl}:
unit-mode: -----> {co}:
line-status-trap-enable: -> {disabled}:
admin-up-line-alarm: -----> {disabled}:
dsl-dpboprofile: -----> {0}: 3
dsl-dpbofallbackprofile: -> {0}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Once DPBO is applied to the interface any new virtual interfaces (bridge or IP) will use the DPBO settings. For existing links, DPBO is applied when a line is bounced (using the **port bounce** command) or when the **dsl-config** profile is updated. Adding or changing any of the DPBO profiles does NOT force a DPBO change.

Applying DPBO to an ADSL2+ interface

Notice that there is a single index which defines the DPBO associated profiles.

- 1 Create a new **dpbo-profile** and configure with the appropriate settings

```
zSH> new dpbo-profile 1

dpbo-profile 3
Please provide the following: [q]uit.
dpbo-name: --> {dpboprof}: ADSL2_DPBO
dpbo-esel: --> {0}: 1275
dpbo-escma: -> {0}: 270
dpbo-escmb: -> {0}: 490
dpbo-escmc: -> {0}: 264
dpbo-fmax: --> {32}: 512
dpbo-fmin: --> {0}: 64
dpbo-mus: ---> {0}: -1110
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

- 2 Create a new **dpbo-epsd** profile and configure with the appropriate settings

```
zSH> new dpbo-epsd 3
```

```
dpbo-epsd 3
```

```
Please provide the following: [q]uit.
```

```
dpbo-epsdfreq1: ---> {0}: 276
```

```
dpbo-epsdlevel1: --> {0}: -500
```

```
dpbo-epsdfreq2: ---> {0}: 2208
```

```
dpbo-epsdlevel2: --> {0}: -500
```

```
dpbo-epsdfreq3: ---> {0}:
```

```
dpbo-epsdlevel3: --> {0}:
```

```
dpbo-epsdfreq4: ---> {0}:
```

```
dpbo-epsdlevel4: --> {0}:
```

```
dpbo-epsdfreq5: ---> {0}:
```

```
dpbo-epsdlevel5: --> {0}:
```

```
dpbo-epsdfreq6: ---> {0}:
```

```
dpbo-epsdlevel6: --> {0}:
```

```
dpbo-epsdfreq7: ---> {0}:
```

```
dpbo-epsdlevel7: --> {0}:
```

```
dpbo-epsdfreq8: ---> {0}:
```

```
dpbo-epsdlevel8: --> {0}:
```

```
dpbo-epsdfreq9: ---> {0}:
```

```
dpbo-epsdlevel9: --> {0}:
```

```
dpbo-epsdfreq10: --> {0}:
```

```
dpbo-epsdlevel10: -> {0}:
```

```
dpbo-epsdfreq11: --> {0}:
```

```
dpbo-epsdlevel11: -> {0}:
```

```
dpbo-epsdfreq12: --> {0}:
```

```
dpbo-epsdlevel12: -> {0}:
```

```
dpbo-epsdfreq13: --> {0}:
```

```
dpbo-epsdlevel13: -> {0}:
```

```
dpbo-epsdfreq14: --> {0}:
```

```
dpbo-epsdlevel14: -> {0}:
```

```
dpbo-epsdfreq15: --> {0}:
```

```
dpbo-epsdlevel15: -> {0}:
```

```
dpbo-epsdfreq16: --> {0}:
```

```
dpbo-epsdlevel16: -> {0}:
```

```
.....
```

```
Save new record? [s]ave, [c]hange or [q]uit: s
```

```
New record saved.
```

- 3** <Optional> Create new **dpbo-psdmask** profile and **dpbo-lfo** profile and configure with the appropriate settings

The **dpbo-psdmask** profile and **dpbo-lfo** profiles are not required. If one or the other is not created a default mask for their properties is not used. The default mask is not needed.

```
zSH> new dpbo-psdmask 1
```

```
dpbo-psdmask 1
```

```
Please provide the following: [q]uit.
```

```

dpbo-psdmaskfreq1: ---> {0}: 276
dpbo-psdmasklevel1: --> {0}: -1100
dpbo-psdmaskfreq2: ---> {0}: 1164
dpbo-psdmasklevel2: --> {0}: -1100
dpbo-psdmaskfreq3: ---> {0}: 1174
dpbo-psdmasklevel3: --> {0}: -400
dpbo-psdmaskfreq4: ---> {0}: 1780
dpbo-psdmasklevel4: --> {0}: -500
dpbo-psdmaskfreq5: ---> {0}: 1786
dpbo-psdmasklevel5: --> {0}: -850
dpbo-psdmaskfreq6: ---> {0}: 2005
dpbo-psdmasklevel6: --> {0}: -850
dpbo-psdmaskfreq7: ---> {0}: 2010
dpbo-psdmasklevel7: --> {0}: -500
dpbo-psdmaskfreq8: ---> {0}: 2163
dpbo-psdmasklevel8: --> {0}: -500
dpbo-psdmaskfreq9: ---> {0}: 2168
dpbo-psdmasklevel9: --> {0}: -850
dpbo-psdmaskfreq10: --> {0}: 2201
dpbo-psdmasklevel10: -> {0}: -850
dpbo-psdmaskfreq11: --> {0}:
dpbo-psdmasklevel11: -> {0}:
dpbo-psdmaskfreq12: --> {0}:
dpbo-psdmasklevel12: -> {0}:
dpbo-psdmaskfreq13: --> {0}:
dpbo-psdmasklevel13: -> {0}:
dpbo-psdmaskfreq14: --> {0}:
dpbo-psdmasklevel14: -> {0}:
dpbo-psdmaskfreq15: --> {0}:
dpbo-psdmasklevel15: -> {0}:
dpbo-psdmaskfreq16: --> {0}:
dpbo-psdmasklevel16: -> {0}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```

4 Apply the DPBO settings to an ADSL2+ interface

```

zSH> update dsl-config 1-10-10-0/adsl

dsl-config 1-10-10-0/adsl
Please provide the following: [q]uit.
line-type: -----> {adsl}:
unit-mode: -----> {co}:
line-status-trap-enable: -> {disabled}:
admin-up-line-alarm: -----> {disabled}:
dsl-dpboprofile: -----> {0}: 1
dsl-dpbofallbackprofile: -> {0}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Once DPBO is applied to the interface any new virtual interfaces (bridge or IP) will use the DPBO settings. For existing links, DPBO is applied when a

line is bounced (using the **port bounce** command) or when the **dsl-config** profile is updated. Adding or changing any of the DPBO profiles does NOT force a DPBO change.

VDSL2 statistics

This chapter describes:

- [View VDSL2 statistics, page 1116](#)
- [View VDSL2 statistics with the -v variable, page 1117](#)
- [Clear VDSL2 counters, page 1119](#)
- [VDSL statistics parameters, page 1119](#)

View VDSL2 statistics

Use the **dslstat** *shelf/slot/port/subport/interface type* command to retrieve or clear DSL statistics for any VDSL port in the system. Entering the **dslstat** command with the -v (verbose) variable retrieves all available statistics.

Viewing VDSL2 statistics

View VDSL2 statistics.

```
zSH> dslstat 1-1-1-0/vdsl
General Stats:
-----
AdminStatus.....UP
LineStatus.....DATA
Line uptime (DD:HH:MM:SS).....0:20:34:09
DslUpLineRate (bitsPerSec).....28183000
DslDownLineRate (bitsPerSec).....86818000
DslMaxAttainableUpLineRate (bitsPerSec).....38366000
DslMaxAttainableDownLineRate (bitsPerSec)....97560000
Out Pkts/Cells/Frags.....21
Out Discards.....0
Out Errors.....0
In Pkts/Cells/Frags.....slots
233
In Discards.....49516
In Errors.....0
DSL Physical Stats:
-----
Actual Transmission connection standard.....VDSL2
Vdsl2CurrentProfile.....g993-2-17a
DslLineSnrMgn (tenths dB).....59
DslLineAtn (tenths dB).....33
DslCurrOutputPwr (tenths dB).....145
LOFS.....0
LOLS.....0
LOSS.....0
ESS.....0
CRC Errors.....0
Inits.....0
```


View VDSL2 stats for vectoring

The **dslstat** command displays statistics for vectoring on VDSL2 cards with vectoring.

View VDSL2 statistics with the -v variable

Using the -v (verbose) variable with the **dslstat** command displays all available statistics.



Note: Statistics in bold indicate Phy-R™ statistics.

View all available VDSL2 statistics

To view all available statistics on VDSL2 enter:

```
zSH> dslstat 1-1-1-0/vdsl -v
General Stats:
-----
AdminStatus.....UP
LineStatus.....DATA
Line uptime (DD:HH:MM:SS).....2:00:15:23
DslUpLineRate (bitsPerSec).....38549000
DslDownLineRate (bitsPerSec).....92932000
DslMaxAttainableUpLineRate (bitsPerSec).....38513000
DslMaxAttainableDownLineRate (bitsPerSec)....94312000
Out Pkts/Cells/Frags.....6702743
Out Discards.....24846
Out Errors.....0
In Pkts/Cells.....3474987
In Discards.....220009
In Errors.....0
DSL Physical Stats:
-----
Actual Transmission connection standard.....VDSL2
Vdsl2CurrentProfile.....g993-2-17a
DslLineSnrMgn (tenths dB).....63
DslLineAtn (tenths dB).....33
DslCurrOutputPwr (tenths dB).....145
LOFS.....30
LOLS.....233
LOSS.....277
ESS.....298
CRC Errors.....21
Inits.....10
near-end statistics:
-----
Loss of Frame Seconds.....0
Loss of Signal Seconds.....277
Loss of Link Seconds.....233
```

```

Severely Errored Seconds.....277
Unavailable Seconds.....277
Retransmitted codewords.....0
Corrected Retransmitted codewords.....0
UnCorrectable Retransmitted codewords.....0
far-end statistics:
-----
Loss of Frame Seconds.....0
Loss of Signal Seconds.....247
Loss of Link Seconds.....44
Severely Errored Seconds.....247
Unavailable Seconds.....247
Retransmitted codewords.....0
Corrected Retransmitted codewords.....0
UnCorrectable Retransmitted codewords.....0
Loss of Power (dying gasps).....44
XTUC PHY Stats:
-----
serialNumber.....121 v10.03.08, 2009-11-17
vendorId.....BDCM 0x4d54
versionNumber.....VE_10_3_8
curSnrMargin (tenths dB).....63
currAtn (tenths dB).....33
currStatus.....NO DEFECT
currOutputPwr (tenths dB).....145
currAttainableRate (bitsPerSec).....94312000
currLineRate (bitsPerSec).....0
XTUC CHAN Stats:
-----
interleaveDelay (tenths milliseconds).....0
crcBlockLength (bytes).....0
currTxRate (bitsPerSec).....92932000
currTxSlowBurstProt.....0
currTxFastFec.....0
XTUR PHY Stats:
-----
serialNumber.....
vendorId.....BDCM 0
versionNumber.....A2pv6C016
curSnrMargin (tenths dB).....63
currAtn (tenths dB).....0
currStatus.....NO DEFECT
currOutputPwr (tenths dB).....76
currAttainableRate (bitsPerSec).....38513000
currLineRate (bitsPerSec).....0
XTUR CHAN Stats:
-----
interleaveDelay (tenths milliseconds).....0
crcBlockLength (bytes).....0
currTxRate (bitsPerSec).....38549000
currTxSlowBurstProt.....0
currTxFastFec.....0

```

Clear VDSL2 counters

Clearing DSL counters

You can clear DSL counters to make identifying the changing statistics easier to read.

- 1 Clear the statistics using the **dslstat clear** command

```
zSH> dslstat clear 1-1-1-0/vdsl
```

- 2 View the changes.

For reference the **dslstat** command (see [Viewing VDSL2 statistics on page 1116](#)) shows the statistics prior to clearing the statistics. Statistics which are cleared by the **dslstat clear** command are in *bold*.

```
zSH> dslstat 1-1-1-0/vdsl
General Stats:
-----
AdminStatus.....UP
LineStatus.....HANDSHAKE
DslUpLineRate (bitsPerSec).....30869000
DslDownLineRate (bitsPerSec).....85406000
DslMaxAttainableUpLineRate (bitsPerSec).....41570000
DslMaxAttainableDownLineRate (bitsPerSec)....97976000
Out Pkts/Cells.....0
Out Discards.....0
Out Errors.....0
In Pkts/Cells.....0
In Discards.....0
In Errors.....0
DSL Physical Stats:
-----
DslLineSnrMgn (tenths dB).....0
DslLineAtn (tenths dB).....0
DslCurrOutputPwr (tenths dB).....0
LOFS.....0
LOLS.....15
LOSS.....15
ESS.....15
CRC Errors.....0
Inits.....0
```

VDSL statistics parameters

[Table 123](#) defines the statistics displayed in the **dslstat** command for an VDSL line.

Table 123: VDSL2 statistics

Statistic	Description
AdminStatus	Administrative status of the port: Values: Up Interface is ready to pass packets. Down Interface is unable to pass packets. Testing Interface is in a special testing state and is unable to pass packets.
LineStatus	Line status provides information about the VDSL2 link. Values for a single VDSL2 line: ACT — the line currently has link and can pass traffic in both directions OOS — the line does not have link TRAFFIC DISABLE — The line currently has link but not underlying VDSL2 protocol; traffic will not pass.
Line uptime (DD:HH:MM:SS)	How long the interface has been up in dd hh mm (day, hour, minute, second) format.
DslUpLineRate (bitsPerSec)	Displays the DSL upstream (customer premise > central office) line rate on this interface.
DslDownLineRate (bitsPerSec)	Displays the DSL downstream (central office > customer premise) line rate on this interface.
DslMaxAttainableUpLineRate (bitsPerSec)	Displays the maximum line rate that can be supported on this line in the upstream direction.
DslMaxAttainableDownLineRate (bitsPerSec)	Displays the maximum line rate that can be supported on this line in the downstream direction.
Out Pkts/Cells	Number of transmitted packets/cells
Out Discards	Number of transmission discards.
Out Errors	Number of transmission errors.
In Pkts/Cells	Number of transmitted packets/cells
In Discards	Number of received discards.
In Errors	Number of receive errors.
VDSL2 Physical Stats:	
Actual Transmission connection standard	Indicates the maximum line rate that can be supported on this line in the downstream direction.
Vdsl2CurrentProfile	The VDSL2 standard to be used for the line.
DslLineSnrMgn (tenths dB)	DSL Line Signal to Noise Ratio Margin — The strength of the DSL signal relative to the noise on line.

Table 123: VDSL2 statistics (Continued)

Statistic	Description
DslLineAtn (tenths dB)	DSL Line Attenuation — Measure of the signal degradation between the VDSL2 port and the modem.
DslCurrOutputPwr (tenths dB)	Not currently used.
LOFS	Number of Loss of Frame Seconds.
LOLS	Number of Loss of Line Seconds.
LOSS	Number of Loss of Signal Seconds.
ESS	Number of errored seconds (the number of one-second intervals containing one or more CRC anomalies or one or more LoS or Sef defects) that has been reported in the current 15-minute interval.
CRC Errors	Cyclic Redundancy Check Errors — CRC checks for transmission errors. The CRC code is computed from the data in the message. If the data is altered the CRC computation will not be in agreement with the data.
Initis	Number of line initialization attempts, including both successful and failed attempts.
near-end (CO) statistics:	
Loss of Frame Seconds	Count of seconds during this interval that there was Loss of Framing.
Loss of Signal Seconds	Count of seconds during this interval that there was Loss of Signal.
Loss of Link Seconds	Count of seconds during this interval that there was Loss of Link.
Severely Errored Seconds	Count of Severely Errored Seconds during this interval.
Unavailable Seconds	Count of Unavailable Seconds during this interval.
Retransmitted codewords	Retransmitted Codewords.
Corrected Retransmitted codewords	Retransmitted corrected Codewords.
UnCorrectable Retransmitted codewords	Retransmitted uncorrectable Codewords.
far-end statistics:	
Loss of Frame Seconds	Count of seconds during this interval that there was Loss of Framing.
Loss of Signal Seconds	Count of seconds during this interval that there was Loss of Signal.
Loss of Link Seconds	Count of seconds during this interval that there was Loss of Link.
Severely Errored Seconds	Count of Severely Errored Seconds during this interval.
Unavailable Seconds	Count of Unavailable Seconds (UAS) during this interval.
Retransmitted codewords	Retransmitted Codewords.
Corrected Retransmitted codewords	Retransmitted corrected Codewords.

Table 123: VDSL2 statistics (Continued)

Statistic	Description
UnCorrectable Retransmitted codewords	Retransmitted uncorrectable Codewords.
Loss of Power (dying gasps)	Count of Loss of Power (LPR) Seconds during this interval.
XTUC PHY Stats:	
serialNumber	The vendor specific string that identifies the vendor equipment.
vendorID	The vendor ID code is a copy of the binary vendor identification field expressed as readable characters in hexadecimal notation.
versionNumber	The vendor specific version number sent by this Vtu as part of the initialization messages. It is a copy of the binary version number field expressed as readable characters in hexadecimal notation.
curSnrMargin (tenths dB)	Noise Margin as seen by this Vtu with respect to its received signal in 0.25dB. The effective range is -31.75 to +31.75 dB.
currAtn (tenths dB)	Measured difference in the total power transmitted by the peer Vtu and the total power received by this Vtu. The effective range is 0 to +63.75 dB.
currStatus	Indicates current state of the Vtu line. This is a bit-map of possible conditions. The various bit positions are: noDefect There are no defects on the line. lossOfFraming Vtu failure due to not receiving Frame. lossOfSignal Vtu failure due to not receiving Signal. lossOfPower Vtu failure due to loss of power. lossOfSignalQuality Loss of Signal Quality is declared when the Noise Margin falls below the Minimum Noise Margin, or the bit-error-rate exceeds 10^{-7} . lossOfLink Vtu failure due to inability to link with peer Vtu. Set whenever the transceiver is in the 'Warm Start' state. dataInitFailure Vtu failure during initialization due to bit errors corrupting. configInitFailure Vtu failure during initialization due to peer Vtu not able to support requested configuration. protocolInitFailure Vtu failure during initialization due to incompatible protocol used by the peer Vtu. noPeerVtuPresent Vtu failure during initialization due to no activation sequence detected from peer Vtu.
currOutputPwr (tenths dB)	Measured total output power transmitted by this VTU. This is the measurement that was reported during the last activation sequence.

Table 123: VDSL2 statistics (Continued)

Statistic	Description
currAttainableRate (bitsPerSec)	Indicates the maximum currently attainable data rate in steps of 1000 bits/second by the Vtu. This value will be equal to or greater than vdslPhysCurrLineRate. Note that for SCM, the minimum and maximum data rates are equal. Note: 1 kbps = 1000 bps.
currLineRate (bitsPerSec)	Indicates the current data rate in steps of 1000 bits/second by the Vtu. This value will be less than or equal to vdslPhysCurrAttainableRate. Note: 1 kbps = 1000 bps
XTUC CHAN Stats:	
interleaveDelay (tenths milliseconds)	Interleave Delay for this channel. Interleave delay applies only to the interleave (slow) channel and defines the mapping (relative spacing) between subsequent input bytes at the interleave input and their placement in the bit stream at the interleave output. Larger numbers provide greater separation between consecutive input bytes in the output bit stream allowing for improved impulse noise immunity at the expense of payload latency. In the case where the ifType is fast(125), return a value of zero.
crcBlockLength (bytes)	Indicates the length of the channel data-block on which the CRC operates.
currTxRate (bitsPerSec)	Actual transmit data rate on this channel. Note: 1 kbps = 1000 bps.
currTxSlowBurstProt	Actual level of impulse noise (burst) protection for an interleaved (slow) channel. This parameter is not applicable to fast channels. For fast channels, a value of zero shall be returned.
currTxFastFec	Actual Forward Error Correction (FEC) redundancy related overhead for a fast channel. This parameter is not applicable to an interleaved (slow) channel. For interleaved channels, a value of zero shall be returned.
XTUR PHY Stats:	
serialNumber	The vendor specific string that identifies the vendor equipment.
vendorId	The vendor ID code is a copy of the binary vendor identification field expressed as readable characters in hexadecimal notation.
versionNumber	The vendor specific version number sent by this Vtu as part of the initialization messages. It is a copy of the binary version number field expressed as readable characters in hexadecimal notation.
curSnrMargin (tenths dB)	Noise Margin as seen by this Vtu with respect to its received signal in 0.25dB. The effective range is -31.75 to +31.75 dB.

Table 123: VDSL2 statistics (Continued)

Statistic	Description
currAtn (tenths dB)	Measured difference in the total power transmitted by the peer Vtu and the total power received by this Vtu. The effective range is 0 to +63.75 dB.
currStatus.	Indicates current state of the Vtu line. This is a bit-map of possible conditions. The various bit positions are: noDefect There are no defects on the line. lossOfFraming Vtu failure due to not receiving. lossOfSignal Vtu failure due to not receiving. lossOfPower Vtu failure due to loss of power. lossOfSignalQuality Loss of Signal Quality is declared when the Noise Margin falls below the Minimum Noise Margin, or the bit-error-rate exceeds 10^{-7}. lossOfLink Vtu failure due to inability to link with peer Vtu. Set whenever the transceiver is in the 'Warm Start' state. dataInitFailure Vtu failure during initialization due to bit errors corrupting. configInitFailure Vtu failure during initialization due to peer Vtu not able to support requested configuration. protocolInitFailure Vtu failure during initialization due to incompatible protocol used by the peer Vtu. noPeerVtuPresent Vtu failure during initialization due to no activation sequence detected from peer Vtu.
currOutputPwr (tenths dB)	Measured total output power transmitted by this VTU. This is the measurement that was reported during the last activation sequence.
currAttainableRate (bitsPerSec)	Indicates the maximum currently attainable data rate in steps of 1000 bits/second by the Vtu. This value will be equal to or greater than vdslPhysCurrLineRate. Note that for SCM, the minimum and maximum data rates are equal. Note: 1 kbps = 1000 bps
currLineRate (bitsPerSec)	Indicates the current data rate in steps of 1000 bits/second by the Vtu. This value will be less than or equal to vdslPhysCurrAttainableRate. Note: 1 kbps = 1000 bps
XTUR CHAN Stats:	

Table 123: VDSL2 statistics (Continued)

Statistic	Description
interleaveDelay (tenths milliseconds)	Interleave Delay for this channel. Interleave delay applies only to the interleave (slow) channel and defines the mapping (relative spacing) between subsequent input bytes at the interleave input and their placement in the bit stream at the interleave output. Larger numbers provide greater separation between consecutive input bytes in the output bit stream allowing for improved impulse noise immunity at the expense of payload latency. In the case where the ifType is fast(125), return a value of zero.
crcBlockLength (bytes)	Indicates the length of the channel data-block on which the CRC operates.
currTxRate (bitsPerSec)	Actual transmit data rate on this channel. Note: 1 kbps = 1000 bps.
currTxSlowBurstProt	Actual level of impulse noise (burst) protection for an interleaved (slow) channel. This parameter is not applicable to fast channels. For fast channels, a value of zero shall be returned.
currTxFastFec	Actual Forward Error Correction (FEC) redundancy related overhead for a fast channel. This parameter is not applicable to an interleaved (slow) channel. For interleaved channels, a value of zero shall be returned.

VDSL2 24-port card pinouts

VDSL2 24-port cards use standard RJ-21X pinouts. [Table 124](#) lists the port pinouts.

Table 124: VDSL2 24-port card pinouts

Pin	Function	Pin	Function
1	Channel 1 ring	26	Channel 1 tip
2	Channel 2 ring	27	Channel 2 tip
3	Channel 3 ring	28	Channel 3 tip
4	Channel 4 ring	29	Channel 4 tip
5	Channel 5 ring	30	Channel 5 tip
6	Channel 6 ring	31	Channel 6 tip
7	Channel 7 ring	32	Channel 7 tip
8	Channel 8 ring	33	Channel 8 tip
9	Channel 9 ring	34	Channel 9 tip
10	Channel 10 ring	35	Channel 10 tip
11	Channel 11 ring	36	Channel 11 tip
12	Channel 12 ring	37	Channel 12 tip
13	Channel 13 ring	38	Channel 13 tip
14	Channel 14 ring	39	Channel 14 tip
15	Channel 15 ring	40	Channel 15 tip
16	Channel 16 ring	41	Channel 16 tip
17	Channel 17 ring	42	Channel 17 tip
18	Channel 18 ring	43	Channel 18 tip
19	Channel 19 ring	44	Channel 19 tip
20	Channel 20 ring	45	Channel 20 tip
21	Channel 21 ring	46	Channel 21 tip
22	Channel 22 ring	47	Channel 22 tip
23	Channel 23 ring	48	Channel 23 tip
24	Channel 24 ring	49	Channel 24 tip
25	Not used	50	Not used

VDSL2 48-port card pinouts

This section describes the VDSL2 48-port pinouts for the MXK-VDSL2-BCM-17a-48-V card.

[Figure 149](#) and [Figure 148](#) show the location of pinouts on the card connectors.

Figure 148: VDSL2 48-port card connector pinouts

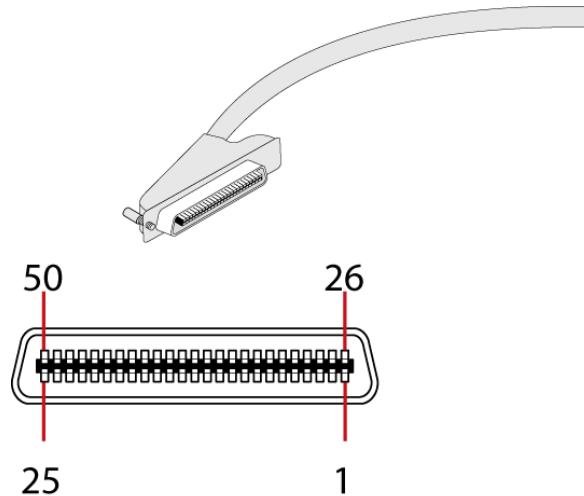


Figure 149: VDSL2 48-port card connector pinouts

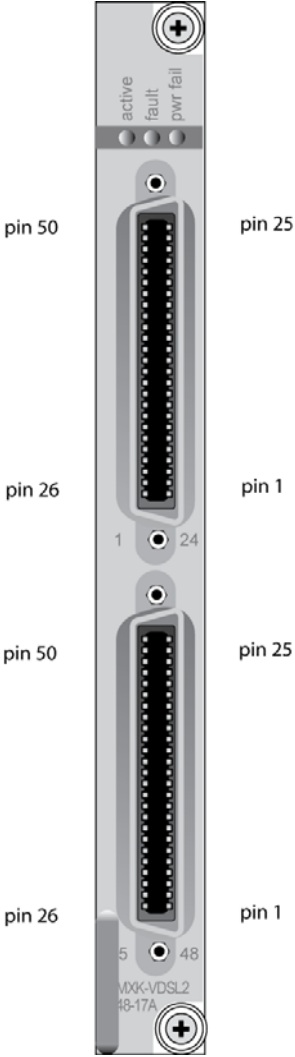


Table 125: 48-port VDSL2 to 50-pin connector and cable

Pair	Signal	Color	From	To
1	Ring	BLU/WHT	J7-1	J7-26
	Tip	WHT/BLU	J7-26	J7-1
2	Ring	ORG/WHT	J7-2	J7-27
	Tip	WHT/ORG	J7-27	J7-2
3	Ring	GRN/WHT	J7-3	J7-28
	Tip	WHT/GRN	J7-28	J7-3
4	Ring	BRN/WHT	J7-4	J7-29
	Tip	WHT/BRN	J7-29	J7-4
5	Ring	GRY/WHT	J7-5	J7-30

Table 125: 48-port VDSL2 to 50-pin connector and cable (Continued)

Pair	Signal	Color	From	To
6	Tip	WHT/GRY	J7-30	J7-5
	Ring	BLU/RED	J7-6	J7-31
7	Tip	RED/BLU	J7-31	J7-6
	Ring	ORG/RED	J7-7	J7-32
8	Tip	RED/ORG	J7-32	J7-7
	Ring	GRN/RED	J7-8	J7-33
9	Tip	RED/GRN	J7-33	J7-8
	Ring	BRN/RED	J7-9	J7-34
10	Tip	RED/BRN	J7-34	J7-9
	Ring	GRY/RED	J7-10	J7-35
11	Tip	RED/GRY	J7-35	J7-10
	Ring	BLU/BLK	J7-11	J7-36
12	Tip	BLK/BLU	J7-36	J7-11
	Ring	ORG/BLK	J7-12	J7-37
13	Tip	BLK/ORG	J7-37	J7-12
	Ring	GRN/BLK	J7-13	J7-38
14	Tip	BLK/GRN	J7-38	J7-13
	Ring	BRN/BLK	J7-14	J7-39
15	Tip	BLK/BRN	J7-39	J7-14
	Ring	GRY/BLK	J7-15	J7-40
16	Tip	BLK/GRY	J7-40	J7-15
	Ring	BLU/YEL	J7-16	J7-41
17	Tip	YEL/BLU	J7-41	J7-16
	Ring	ORG/YEL	J7-17	J7-42
18	Tip	YEL/ORG	J7-42	J7-17
	Ring	GRN/YEL	J7-18	J7-43
19	Tip	YEL/GRN	J7-43	J7-18
	Ring	BRN/YEL	J7-19	J7-44
20	Tip	YEL/BRN	J7-44	J7-19
	Ring	GRY/YEL	J7-20	J7-45

Table 125: 48-port VDSL2 to 50-pin connector and cable (Continued)

Pair	Signal	Color	From	To
	Tip	YEL/GRY	J7-45	J7-20
21	Ring	BLU/VIO	J7-21	J7-46
	Tip	VIO/BLU	J7-46	J7-21
22	Ring	ORG/VIO	J7-22	J7-47
	Tip	VIO/ORG	J7-47	J7-22
23	Ring	GRN/VIO	J7-23	J7-48
	Tip	VIO/GRN	J7-48	J7-23
24	Ring	BRN/VIO	J7-24	J7-49
	Tip	VIO/BRN	J7-49	J7-24
25 (spare)	Ring	GRY/VIO	J7-25	J7-50
	Tip	VIO/GRY	J7-50	J7-25

12

MXK ACTIVE ETHERNET CARDS

This chapter describes the MXK 20-port Active Ethernet dual-slot card and Active Ethernet single-slot cards:

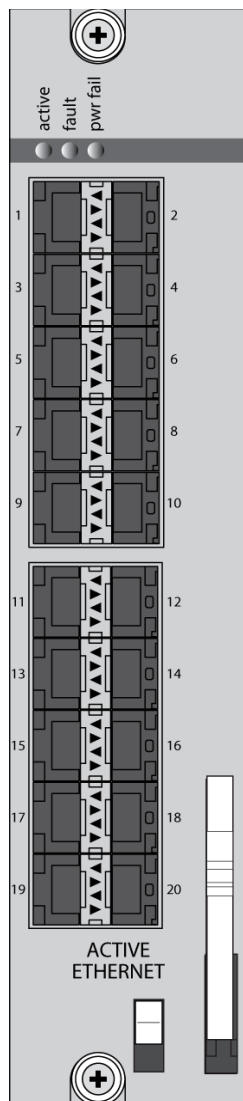
- [20-port Active Ethernet dual-slot card, page 1131](#)
- [20-port Active Ethernet single-slot card, page 1137](#)
- [20-port Active Ethernet single-slot card with C-SFP support, page 1142](#)
- [10-port Active Ethernet single-slot card with 2X10G-8XGE, page 1147](#)
- [Displaying and updating Ethernet interfaces, page 1152](#)
- [Small form factor pluggables, page 1154](#)
- [Ethernet redundancy, page 1154](#)
- [Port redundancy on Active Ethernet line cards, page 1164](#)
- [Default Ethernet alarms on line card Minor, page 1165](#)
- [Settable alarm severity for Ethernet ports, page 1165](#)
- [Enhanced Ethernet port statistics, page 1168](#)

20-port Active Ethernet dual-slot card

This section describes the MXK 20-port dual-slot Active Ethernet card and Active Ethernet card configuration:

- [Active Ethernet dual-slot card overview, page 1132](#)
- [Active Ethernet dual-slot card specifications, page 1133](#)
- [Active Ethernet dual-slot card configuration, page 1133](#)
- [View additional card and system information, page 1135](#)

Active Ethernet dual-slot card overview



The MXK Active Ethernet dual-slot line card, MXK-AEX20-FE/GE-2S, is a two slot card that supports Ethernet traffic over 20 ports that provide either 100/1000 Base-T, fiber 100FX or 1 Gigabit Ethernet interfaces to support distances as high as 80km depending on the SFPs used. The Active Ethernet card is also interoperable third party Active Ethernet devices.

The Active Ethernet card supports Layer 2 bridging functions, Layer 2 security functions, Layer 3 routing functions and the Zhone Multimedia Traffic Management functionality (MTM).

This card supports non-redundant GigE connections to subtended SLMS devices.

Active Ethernet dual-slot card specifications

[Table 126](#) provides the Active Ethernet MXK-AEX20-FE/GE-2S card specifications.

Table 126: Active Ethernet MXK-AEX20-FE/GE-2S card specifications

Specification	Description
Size	2 slot
Density	20 GigE ports
Physical interfaces	100/1000 Ethernet ports with SFPs. The optical interfaces are class 1 Laser International Safety Standard IEC 825 compliant 20 Gigabit Ethernet ports with SFPs. The SFPs can be twisted pair 1000baseT or fiber (SX, LX or ZX). See Small form factor pluggables on page 1154 .
Standards supported	IEEE 802.3 IEEE 802.1 Q/P IEEE 802.1 AD (Q in Q)
Power consumption	52.3 W

Active Ethernet dual-slot card configuration

Each card installed in the system must have a **card-profile**. The line card type determines the parameter settings in the **card-profile** and the software image for the card. Performing a **card add** <slot #> automatically creates the **card-profile** for the card with the correct software image and settings.

[Table 127](#) provides the type and software image for the Active Ethernet cards on the MXK.

Table 127: MXK card type for Active Ethernet

Card	Type	Name of software image
MXK-AEX20-FE/GE-2S	10200	<i>mxlc20ae.bin</i>

Configuring an Active Ethernet card

To configure an Active Ethernet card on the system:

- 1 Install the Active Ethernet card in the desired line card slot.
- 2 Verify the card by entering **slots**:

```
zSH> slots
MXK 819
Uplinks
a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
```

```

b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
4: MXK 4 PORT GPON (RUNNING+TRAFFIC)
13: MXK 20 ACT ETH (RUNNING)

```

3 Create a **card-profile** for the card:

```
zSH> card add 13
```

4 Verify the **card-profile** for the Active Ethernet card:

```

zSH> get card-profile 1/13/10200
card-profile 1/13/10200
sw-file-name: -----> {mxlc20ae.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {unknowntype}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}

```

5 Connect the line-side cables to the SFP connectors on the Active Ethernet card.

Verifying the line card installation

After you save the **card-profile** record, the line card in that slot resets and begins downloading the software image from the flash card. This could take a few moments.

When the card has finished loading, a log message similar to the following is displayed (if logging is enabled):

```
zSH> Card in slot slot-number changed state to RUNNING
```

1 To view the status of all the cards enter **slots**:

```

zSH> slots
Uplinks
a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
4: MXK 4 PORT GPON (RUNNING)
13: MXK 20 ACT ETH (NOT_PROV)

```

- 2 To view card information including the state of the card and how long the card has been running you enter **slots** and specify the slot number of the card:

```
zSH> slots 13
Type           : MXK 20 ACT ETH
Card Version    : 800-02316-01-A
EEPROM Version  : 1
Serial #       : 1769100
CLEI Code      : No CLEI
Card-Profile ID : 1/13/10200
Shelf          : 1
Slot           : 13
ROM Version     : MXK 1.16.0.128
Software Version: MXK 1.16.1.128
State          : RUNNING
Mode           : FUNCTIONAL
Heartbeat check : enabled
Longest hbeat   : 80
Fault reset     : enabled
Uptime         : 19 days, 3 hours, 26 minutes
```

View additional card and system information

View the EPROM version of the card:

```
zSH> eeshow card 13
EEPROM contents: for slot 13
EEPROM_ID      : 00 -- CARD
Version        : 01
Size           : 054
CardType       : 10200 -- MXLC20AE
CardVersion    : 800-02316-01-A
SerialNum      : 01769100
ShelfNumber    : 00001
CLEI Code      : No CLEI
Cksum          : 0x4EB4
```

View the ROM version of the card:

```
zSH> romversion 13
MXK 1.16.0.128
Feb  9 2009, 20:10:55
```

Verify the existence of a daughter card on the system:

```
zSH> eeshow 1d
EEPROM contents: for slot 30
EEPROM_ID      : 01 -- 1DAUGHTER
Version        : 01
Size           : 022
CardType       : 10100 -- MXUP2TG8G
CardVersion    : 800-02482-01-B
SerialNum      : 01768971
```

```
ShelfNumber : 00001  
CLEI Code   : No CLEI  
Cksum       : 0xFA1F
```

View the version of the system software:

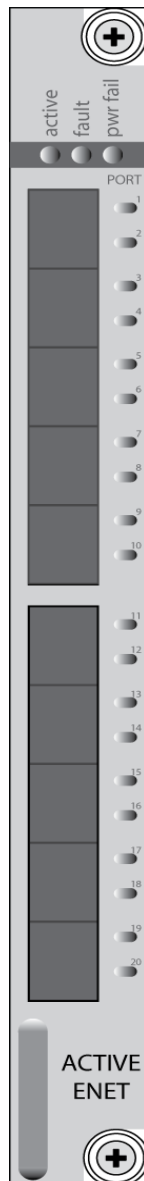
```
zSH> swversion  
Zhone mxUp2Tg8g software version MXK 1.16.1.128
```

20-port Active Ethernet single-slot card

This section describes the single-slot MXK 20-port Active Ethernet card and Active Ethernet card configuration:

- [Active Ethernet single-slot card overview, page 1137](#)
- [Active Ethernet single-slot card specifications, page 1138](#)
- [Active Ethernet single-slot card configuration, page 1138](#)
- [View additional card and system information, page 1140](#)

Active Ethernet single-slot card overview



The MXK Active Ethernet line card, MXK-AEX20-FE/GE, is a single slot card that supports Ethernet traffic over 10 ports that provide either 100/1000 Base-T, fiber 100FX or 1 Gigabit Ethernet interfaces to support distances as high as 80km depending on the SFPs used. The Active Ethernet card is also interoperable third party Active Ethernet devices.

The Active Ethernet card supports Layer 2 bridging functions, Layer 2 security functions, Layer 3 routing functions and the Zhone Multimedia Traffic Management functionality (MTM).

This card supports non-redundant GigE connections to subtended SLMS devices.

Active Ethernet single-slot card specifications

[Table 126](#) provides the Active Ethernet card specifications.

Table 128: Active Ethernet card specifications

Specification	Description
Size	1 slot
Density	20 GigE ports
Physical interfaces	100/1000 Ethernet ports with SFPs. The optical interfaces are class 1 Laser International Safety Standard IEC 825 compliant 20 Gigabit Ethernet ports with SFPs. The SFPs can be twisted pair 1000baseT or fiber (SX, LX or ZX). See Chapter 18, Small Form Factor Pluggable (SFP) Connectors , on page 1117.
Standards supported	IEEE 802.3 IEEE 802.1 Q/P IEEE 802.1 AD (Q in Q)
Power consumption	52.3 W

Active Ethernet single-slot card configuration

Each card installed in the system must have a **card-profile**. The line card type determines the parameter settings in the **card-profile** and the software image for the card. Performing a **card add** <slot #> automatically creates the **card-profile** for the card with the correct software image and settings.

[Table 127](#) shows the type and software image for the Active Ethernet card on the MXK.

Table 129: MXK card type for Active Ethernet

Card	Type	Name of software image
MXK-AEX20-FE/GE	10207	<i>mxlc20ae1s.bin</i>

Adding Active Ethernet cards

To add an Active Ethernet card to the system:

- 1 Install the Active Ethernet card in the desired line card slot.
- 2 Create a **card-profile** for the card:

```
zSH> card add 12
```

After performing a **card add** in a slot, the slot resets and begins downloading the software image from the flash card. This could take a few moments.

When the card has finished loading, a log message similar to the following is displayed (if logging is enabled):

```
zSH> Card in slot slot-number changed state to RUNNING
```

- 3 Verify the card by entering **slots**:

```
zSH> slots
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
  7: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
  9: MXK 20 ACT ETH (RUNNING)
 11: MXK 8 PORT GPON (RUNNING)
 12: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
 13: MXK 20 ACT ETH (RUNNING)
```

- 4 Verify the **card-profile** for the Active Ethernet card:

```
zSH> get card-profile 1/12/10207
card-profile 1/12/10207
sw-file-name: -----> {mxlc20ae1s.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ---> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {unknowntype}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
```

- 5 View card information including the state of the card and how long the card has been running:

```
zSH> slots 12
Type           : MXK 20 ACT ETH SINGLE SLOT
Card Version   : 800-02703-01-Z
EEPROM Version : 1
Serial #       : 9999006
CLEI Code      : PROTO06A08
Card-Profile ID : 1/12/10207
Shelf          : 1
Slot           : 12
ROM Version    : development
Software Version: MXK 1.16.1.211
State          : RUNNING
Mode           : FUNCTIONAL
Heartbeat check : enabled
Longest hbeat  : 348
Fault reset    : enabled
Uptime         : 5 days, 3 hours, 18 minutes
```

- 6 Connect the line-side cables to the SFP connectors on the Active Ethernet card.

View additional card and system information

View the EPROM version of the card:

```
zSH> eeshow card 12
EEPROM contents: for slot 12
EEPROM_ID      : 00 -- CARD
Version        : 01
Size           : 054
CardType        : 10207 -- MXLC20AE_SINGLE_SLOT
CardVersion     : 800-02703-01-Z
SerialNum       : 09999006
ShelfNumber     : 00001
CLEI Code       : PROTO06A08
Cksum           : 0x7B81
```

View the ROM version of the card:

```
zSH> romversion 12
development
Feb  9 2009, 09:01:01
```

Verify the existence of a daughter card on the system:

```
zSH> eeshow 1d
EEPROM contents: for slot 30
EEPROM_ID      : 01 -- 1DAUGHTER
Version        : 01
Size           : 022
CardType        : 10100 -- MXUP2TG8G
CardVersion     : 800-02482-01-D
SerialNum       : 01762719
ShelfNumber     : 00001
```


CLEI Code : No CLEI
Cksum : 0x1289

Verify the existence of a second daughter card by entering **eeshow 2d**.

View the version of the system software:

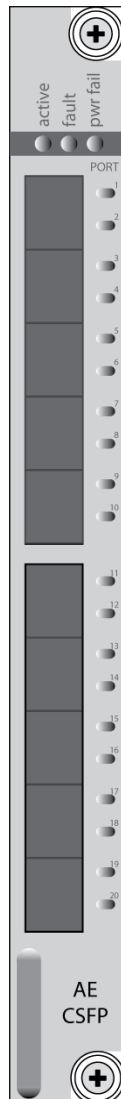
```
zSH> swversion  
Zhone mxUp2Tg8g software version MXK 1.16.2.127
```

20-port Active Ethernet single-slot card with C-SFP support

This section describes the single-slot MXK 20-port Active Ethernet card with compact SFP support and Active Ethernet card configuration:

- [Active Ethernet single-slot card with compact SFP support overview, page 1142](#)
- [Active Ethernet single-slot card with compact SFP support specifications, page 1143](#)
- [Active Ethernet single-slot card with compact SFP support configuration, page 1143](#)
- [View additional card and system information, page 1145](#)

Active Ethernet single-slot card with compact SFP support overview



The MXK Active Ethernet line card, MXK-AEX20-FE/GE-CSFP, is a single slot card that supports multiple subscribers on a single SFP cage through the use of SFPs of type CSFP option 2 with two bi-directional transceivers. This Active Ethernet card also supports single channel SFPs and dual bi-directional (bi-di) SFPs.

The Active Ethernet card supports Layer 2 bridging functions, Layer 2 security functions, Layer 3 routing functions and the Zhone Multimedia Traffic Management functionality (MTM). This card supports non-redundant GigE connections to subtended SLMS devices.

Active Ethernet single-slot card with compact SFP support specifications

[Table 130](#) provides the Active Ethernet single-slot card with C-SFP support specifications.

Table 130: Active Ethernet card with C-SFP support specifications

Specification	Description
Size	1 slot
Density	20 GigE ports
Physical interfaces	100/1000 Ethernet ports with SFPs. The optical interfaces are class 1 Laser International Safety Standard IEC 825 compliant 20 Gigabit Ethernet ports with SFPs. The SFPs can be twisted pair 1000baseT or fiber (SX, LX or ZX). See Chapter 18, Small Form Factor Pluggable (SFP) Connectors , on page 1117.
Standards supported	IEEE 802.3 IEEE 802.1 Q/P IEEE 802.1 AD (Q in Q)
Power consumption	

Active Ethernet single-slot card with compact SFP support configuration

Each card installed in the system must have a **card-profile**. The line card type determines the parameter settings in the **card-profile** and the software image for the card. Performing a **card add <slot #>** automatically creates the **card-profile** for the card with the correct software image and settings.

Table 127 shows the type and software image for the Active Ethernet card on the MXK.

Table 131: MXK card type for Active Ethernet with C-SFP support

Card	Type	Name of software image
MXK-AEX20-FE/GE-CSFP	10216	<i>mxlc20ae1scsfp.bin</i>

Adding Active Ethernet cards

To add an Active Ethernet card to the system:

- 1 Install the Active Ethernet card in the desired line card slot.
- 2 Create a **card-profile** for the card:

```
zSH> card add 6
```

After performing a **card add** in a slot, the slot resets and begins downloading the software image from the flash card.

When the card has finished loading, a log message similar to the following is displayed (if logging is enabled):

```
zSH> Card in slot slot-number changed state to RUNNING
```

- 3 Verify the card by entering **slots**:

```
zSH> slots
MXK 819
Uplinks
  a:*MXK FOUR GIGE (RUNNING+TRAFFIC)
  b: MXK FOUR GIGE (RUNNING)
Cards
  1: MXK 20 ACT ETH (RUNNING)
  5: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
  6: MXK 20 ACT ETH SS CSFP (RUNNING)
  14:*TAC ITM RING (RUNNING)
```

- 4 Verify the **card-profile** for the Active Ethernet card:

```
zSH> get card-profile 1/6/10216
card-profile 1/6/10216
sw-file-name: -----> {mxlc20ae1scsfp.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {unknownntype}
```

```

card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}

```

5 View card information including the state of the card and how long the card has been running:

```

zSH> slots 6
MXK 819
Type           : MXK 20 ACT ETH SS CSFP
Card Version    : 800-03142-01-A
EEPROM Version  : 1
Serial #        : 8467569
CLEI Code       : No CLEI
Card-Profile ID : 1/6/10216
Shelf           : 1
Slot            : 6
ROM Version     : MXK 2.2.1.307
Software Version: MXK 2.3.1.005
State           : RUNNING
Mode            : FUNCTIONAL
Heartbeat check : enabled
Heartbeat last  : TUE DEC 13 18:37:56 2011
Heartbeat resp  : 107709
Heartbeat late  : 0
Hbeat seq error : 0
Hbeat longest   : 10
Fault reset     : enabled
Power fault mon : supported
Uptime          : 1 day, 5 hours, 55 minutes

```

6 Connect the line-side cables to the SFP connectors on the card.

View additional card and system information

View the EPROM version of the card:

```

zSH> eeshow card 6
EEPROM contents: for slot 6
EEPROM_ID      : 00 -- CARD
Version        : 01
Size           : 054
CardType       : 10216 -- MXLC20AE_SINGLE_SLOT_CSFP
CardVersion    : 800-03142-01-A
SerialNum      : 08467569
ShelfNumber    : 00001
CLEI Code      : No CLEI
Cksum          : 0x0E3A

```

View the ROM version of the card:

```
zSH> romversion 6  
MXK 2.2.1.307  
Jul 17 2011, 10:53:09
```

View the version of the system software:

```
zSH> swversion  
Zhone mxUp4G software version MXK 2.3.1.005
```

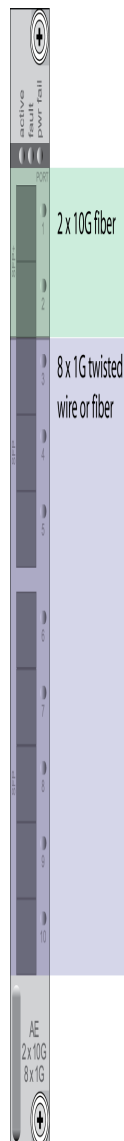
10-port Active Ethernet single-slot card with 2X10G-8XGE

This section describes the MXK 10-port 2X 10G 8X GE Active Ethernet line card:

- [MXK-AE-2X10G-8X1GE line card overview, page 1147](#)
- [MXK-AE-2X10G-8X1GE specifications, page 1148](#)
- [Active Ethernet dual-slot card configuration, page 1133](#)
- [Link aggregation on the MXK-AE-2X10G-8X1GE line card, page 1151](#)
- [SFPs and SFP+s on the MXK-AE-2X10G-8X1GE line card, page 1151](#)

MXK-AE-2X10G-8X1GE line card overview

Figure 150: MXK-AE-2X10G-8X1GE line card



The Active Ethernet MXK-AE-2X10G-8X1GE line card provides dense Active Ethernet trunking with two 10G (1G/10G bps) fiber and eight 1G (100/1G bps) twisted wire or fiber services selectable port by port.

MXK-AE-2X10G-8X1GE specifications

[Table 132](#) provides the Active Ethernet MXK-AE-2X10G-8X1GE line card specifications.

Table 132: Active Ethernet 2X 10G 8X GE card specifications

Specification	Description
Size	1 slot
Density	Two 10 Gig ports (1G/10G) Eight 100/1000 Ethernet ports
Physical interfaces	Two 10 Gig Ethernet ports with SFP+ fiber connections. Eight 100/1000 Ethernet ports with SFPs. The SFPs can be twisted pair 1000baseT or fiber (SX, LX or ZX). The optical interfaces are class 1 Laser International Safety Standard IEC 825 compliant See Chapter 18, Small Form Factor Pluggable (SFP) Connectors , on page 1117.
Standards supported	IEEE 802.3 IEEE 802.1 Q/P IEEE 802.1 AD (Q in Q)
Power consumption	Nominal: 32 W 10G ports: 3.0 W per SFP+ 1GE ports: 1.4 W per SFP Maximum: 49 W

MXK-AE-2X10G-8X1GE configuration

Each card installed in the system must have a **card-profile**. The line card type determines the parameter settings in the **card-profile** and the software image for the card. Performing a **card add <slot #>** automatically creates the **card-profile** for the card with the correct software image and settings.

[Table 127](#) provides the type and software image for the Active Ethernet cards on the MXK.

Table 133: MXK card type for Active Ethernet

Card	Type	Name of software image
MXK-AE-2X10G-8XGE line card	10227	<i>mxlc2tg8gae.bin</i>

Configuring an Active Ethernet card

To configure an Active Ethernet card on the system:

- 1 Install the Active Ethernet card in the desired line card slot.
- 2 Verify the card by entering **slots**:

```
zSH> slots
```

```
MXK 823
```

```
Uplinks
```

```
a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
```

```
b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
```

```
Cards
```

```
1: MXK 2 10G 8 1G ACT ETH (RUNNING)
```

```
2: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
```

```
3: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
```

```
4: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
```

```
5: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
```

```
6: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
```

- 3 Create a **card-profile** for the card:

```
zSH> card add 1
```

- 4 Verify the **card-profile** for the Active Ethernet card:

```
zSH> get card-profile 1/1/10227
```

```
card-profile 1/1/10227
```

```
sw-file-name: -----> {mxlc2tg8gae.bin}
```

```
admin-status: -----> {operational}
```

```
upgrade-sw-file-name: ---> {}
```

```
upgrade-vers: -----> {}
```

```
admin-status-enable: ----> {enable}
```

```
sw-upgrade-admin: -----> {reloadcurrrev}
```

```
sw-enable: -----> {true}
```

```
sw-upgrade-enable: -----> {false}
```

```
card-group-id: -----> {0}
```

```
hold-active: -----> {false}
```

```
weight: -----> {nopreference}
```

```
card-line-type: -----> {unknown type}
```

```
card-atm-configuration: -> {not applicable}
```

```
card-line-voltage: -----> {not-used}
```

```
maxvpi-maxvci: -----> {not applicable}
```

```
card-init-string: -----> {}
```

```
wetting-current: -----> {disabled}
```

```
pwe-timing-mode: -----> {none}
```

- 5 Connect the line-side cables to the SFP connectors on the Active Ethernet card.

Verifying the line card installation

After you save the **card-profile** record, the line card in that slot resets and begins downloading the software image from the flash card. This could take a few moments.

When the card has finished loading, a log message similar to the following is displayed (if logging is enabled):

```
zSH> Card in slot slot-number changed state to RUNNING
```

- 1 To view the status of all the cards enter **slots**:

```
zSH> slots
```

```
MXK 823
```

```
Uplinks
```

```
a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
```

```
b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
```

```
Cards
```

```
1: MXK 2 10G 8 1G ACT ETH (RUNNING)
```

```
2: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
```

```
3: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
```

```
4: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
```

```
5: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
```

```
6: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
```

- 2 To view card information including the state of the card and how long the card has been running you enter **slots** and specify the slot number of the card:

```
zSH> slots 1
```

```
MXK 823
```

```
Type : MXK 2 10G 8 1G ACT ETH
```

```
Card Version : 800-03242-01-A
```

```
EEPROM Version : 1
```

```
Serial # : 1234570
```

```
CLEI Code : No CLEI
```

```
Card-Profile ID : 1/1/10227
```

```
Shelf : 1
```

```
Slot : 1
```

```
ROM Version : MXK 2.4.1.238
```

```
Software Version: MXK 2.5.1.122
```

```
State : RUNNING
```

```
Mode : FUNCTIONAL
```

```
Heartbeat check : enabled
```

```
Heartbeat last : WED APR 23 00:03:40 2014
```

```
Heartbeat resp : 12547
```

```
Heartbeat late : 0
```

```
Hbeat seq error : 0
```

```
Hbeat longest : 10
```

```
Fault reset : enabled
```

```
Power fault mon : supported
```

```
Uptime : 3 hours, 29 minutes
```

Link aggregation on the MXK-AE-2X10G-8X1GE line card

Link aggregation is supported on the MXK-AE-2X10G-8X1GE line card. However, link rates *cannot* be mixed.

The two 10 Gig ports cannot be placed in a link aggregation group with any of the eight 1 GE ports even when a 10 Gig port is configured with a 1 GE SFP. See [Chapter 8, Link Aggregation Configuration, on page 589](#) for link aggregation on the MXK.

SFPs and SFP+s on the MXK-AE-2X10G-8X1GE line card



Note: The two 10 Gig Ethernet ports use only fiber-based SFPs. Copper-based SFPs cannot be used.

See [Chapter 19, Small Form Factor Pluggable \(SFP\) Connectors, on page 1579](#) for SFP information.

Displaying and updating Ethernet interfaces

The **list**, **get**, and **update** commands support use of the interface *shelf-slot-port-subport/eth* syntax to facilitate Ethernet port and interface monitoring and configuration.

To list the currently configured Ethernet interfaces, enter the **list ether** command.

```
zSH> list ether
ether 1-a-1-0/eth
ether 1-a-2-0/eth
ether 1-a-3-0/eth
ether 1-a-4-0/eth
ether 1-a-5-0/eth
ether 1-a-6-0/eth
ether 1-a-7-0/eth
ether 1-a-8-0/eth
ether 1-a-9-0/eth
ether 1-a-10-0/eth
ether 1-a-11-0/eth
ether 1-b-1-0/eth
ether 1-b-2-0/eth
ether 1-b-3-0/eth
ether 1-b-4-0/eth
ether 1-b-5-0/eth
ether 1-b-6-0/eth
ether 1-b-7-0/eth
ether 1-b-8-0/eth
ether 1-b-9-0/eth
ether 1-b-10-0/eth
ether 1-b-11-0/eth
ether 1-13-1-0/eth
ether 1-13-2-0/eth
ether 1-13-3-0/eth
ether 1-13-4-0/eth
...
42 entries found.
```

The **list ether** command shows the Ethernet interfaces on each uplink card as well as the Ethernet interfaces on the Active Ethernet card in slot 13.

The **slots** command verifies the location of the cards with Ethernet interfaces:

```
zSH> slots
MXK 819
Uplinks
  a:*MXK FOUR GIGE (RUNNING+TRAFFIC)
  b: MXK FOUR GIGE (RUNNING)
Cards
  1: MXK 20 ACT ETH (RUNNING)
  3: MXK 20 ACT ETH (RUNNING)
  5: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
  6: MXK 20 ACT ETH SS CSFP (RUNNING)
```

```

8: MXK 20 ACT ETH (RUNNING)
11: MXK ADSL-72-A Bonded (RUNNING)
14:*TAC ITM RING (RUNNING)

```

To view an Ethernet interface, enter the **get ether** command followed by the interface/type.



Note: Link aggregation is not supported on Active Ethernet ports.

```

zSH> get ether 1-6-1-0/eth
ether 1-6-1-0/eth
autonegstatus: ----> {enabled}
mauType: ----> {mau1000basetfd}
restart: ----> {norestart}
ifType: ----> {mau1000basetfd}
autonegcap: ----> {b100baseTX+b100baseTXFD+b1000baseT+b1000baseTFD}
remotefault: ----> {noerror}
clksrc: ----> {automatic}
pauseFlowControl: -> {disabled}
aggregationMode: --> {on}
linkStateMirror: --> {0/0/0/0/0}

```

Small form factor pluggables

Zhone Technologies supports a variety of small form factor pluggables (SFPs) and XFPs which are selected depending on the protocol, fiber type and distance requirements. For information and specifications on supported SFPs and XFPs, see [Chapter 18, Small Form Factor Pluggable \(SFP\) Connectors, on page 1117](#).

Ethernet redundancy

The MXK supports Ethernet redundancy specified in the standards specification.

Ethernet redundant ports provide link protection between Ethernet cards on the MXK to subtended devices such as MXKs (see [Figure 151](#)) and MALCs as well as to Layer 2 (L2) switches (see [Figure 152](#)).

For facility protection to downstream 1U devices such as the MX 160, other facility protection configuration such as link aggregation must be used. For a fully redundant system, EAPS can be configured on the uplinks, and downstream from subtended devices.

Figure 151: Active Ethernet line-card redundancy with a subtended MXK

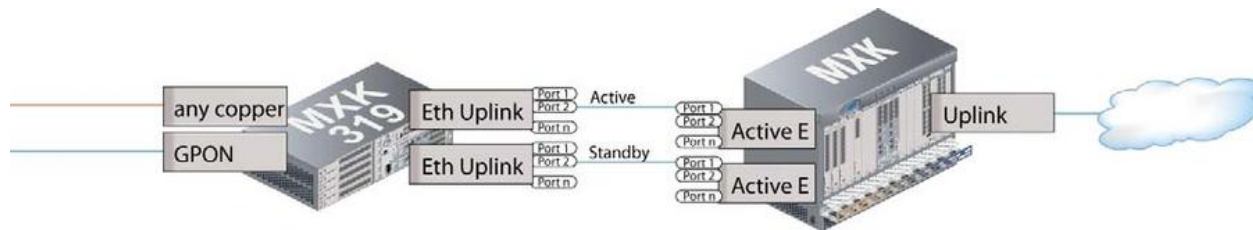
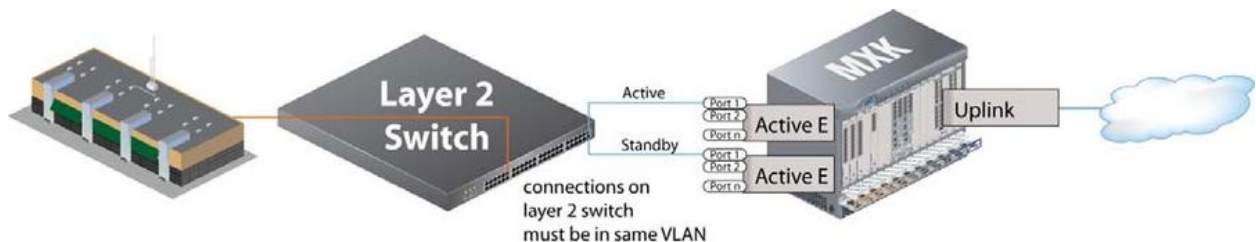


Figure 152: Active Ethernet line-card redundancy to a L2 switch



Ethernet redundancy groups consist of two Ethernet ports. The two Ethernet ports can be on the same or different Ethernet cards. Since it is port level redundancy and not card level redundancy, the port number on one card does not need to match the port number on the second card.

A single Ethernet port cannot be configured in two groups at the same time.

Use the **line-red** command to designate which port is primary or secondary when creating a redundancy group. If you reboot the MXK system (or reboot

the cards which have the redundant ports), the Ethernet port which comes up first and passes traffic becomes the active port.

In a redundancy group, one Ethernet port is always assigned as active and the other as standby. If an active Ethernet port fails, the standby takes over and becomes active. Note that Ethernet redundancy is non-revertive; that is, a previously active Ethernet port which has failed does not become active when the reason for the failover is resolved. The current active port will stay active until that port/line fails, then the standby (if the initial issue was resolved) will once again become the active port.

When a standby port is added to a redundancy group and comes up, the card with the active port copies over the configuration database and routing tables to the standby Ethernet port on the second card. As configuration changes are made to the active port, the standby port is automatically updated.



Note: Create the line redundancy before building interfaces such as bridges. If you add a port with existing interfaces as primary port of the redundant pair, you will need to perform a **slot reboot** on the Ethernet card with the secondary port after adding the redundancy.



Note: All logical interfaces *must* always be created on the *primary* port of the redundant pair.

Create Ethernet line redundancy



Note: Ethernet port redundancy does not work on Ethernet ports that are a part of a link aggregation group.

Configuring Ethernet line redundancy

- 1 Turn off link aggregation capabilities of the Ethernet ports designated for port redundancy.

```
zSH> linkagg update link 1-7-19-0/eth off
```

Warning: this command will similarly update the aggregationMode of every link which is in an aggregation with this link, as well as any redundant peers. Also, changing a link from on or off to active or passive will put the link into an aggregation if it is not in one.
Do you want to continue? [yes] or [no]: **yes**

```
zSH> linkagg update link 1-7-20-0/eth off
```

Warning: this command will similarly update the aggregationMode of every link which is in an aggregation with this link, as well as any redundant peers. Also, changing a link from on or off to active or passive will put the link into an aggregation if it is not in one.
Do you want to continue? [yes] or [no]: **yes**

- 2 Verify existing redundancy with the **line-red show all** command.:

```
zSH> line-red show all
No line redundancy interfaces for system
```

This display shows no redundancy.

You can also display redundancy for a specific port using the **line-red show interface/type** command.

```
zSH> line-red show 1-7-19-0/eth
redundancy status for 1-7-19-0/eth:
      NOREBOOT standbytx ENABLE timeout 0 NONREVERTIVE revert timeout 0
The 1-7-19-0/eth is not part of any redundancy group
```

3 Create line redundancy.

```
zSH> line-red add pri 1-7-19-0/eth sec 1-7-20-0/eth
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
Protection pair has been created. 1-7-19-0/eth is primary and 1-7-20-0/eth
is secondary.
```

4 Verify the line redundancy.



Note: You should wait until redundancy is confirmed before changing any provisioning on the port. Verify the redundancy using one of the following show commands before adding or deleting bridge interfaces or IP interfaces on the Ethernet port.

```
zSH> line-red show all
Line Redundant interfaces for system
Interface-Type      Interface-Name      Oper-State Oper-Status
=====
Primary            1-7-19-0/eth       Active     UP
Secondary          1-7-20-0/eth       Standby    Trfc-Disable
```

You can also show the redundancy by the specific line

```
zSH> line-red show 1-7-19-0/eth
redundancy status for 1-7-19-0/eth:
      NOREBOOT standbytx ENABLE timeout 0 NONREVERTIVE revert timeout 0
Interface-Type      Interface-Name      Oper-State Oper-Status
=====
Primary            1-7-19-0/eth       Active     UP
Secondary          1-7-20-0/eth       Standby    Trfc-Disable
```


Create a downlink bridge interface on redundant Ethernet ports

After creating the redundancy on the Ethernet ports, configure the primary Ethernet port with a bridge interface.

Configuring a downlink bridge interface on a redundant port

- 1 Create a downlink bridge interface on the primary port of the redundant pair.
 - a View the line redundancy.

```
zSH> line-red show 1-7-19-0/eth
redundancy status for 1-7-19-0/eth:
      NOREBOOT standbytx ENABLE timeout 0 NONREVERTIVE revert timeout 0
Interface-Type      Interface-Name      Oper-State Oper-Status
=====
Primary            1-7-19-0/eth        Active      UP
Secondary          1-7-20-0/eth        Standby     Trfc-Disable
```

- b Create the downlink bridge on the primary port.

```
zSH> bridge add 1-7-19-0/eth downlink vlan 100 tagged
Adding bridge on 1-7-19-0/eth
Created bridge-interface-record 1-7-19-0-eth-100/bridge
```

- c Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN      VLAN/SLAN  Bridge                               St  Table Data
-----
dwn      Tagged 100    1-7-19-0-eth-100/bridge             UP
1 bridges displayed
```

- 2 Test line redundancy.

- a Bounce the port.

```
zSH> port bounce 1-7-19-0/eth
1-7-19-0/eth set to admin state DOWN
1-7-19-0/eth set to admin state UP
```

- b Show the line redundancy

```
zSH> line-red show 1-7-19-0/eth
redundancy status for 1-7-19-0/eth:
      NOREBOOT standbytx ENABLE timeout 0 NONREVERTIVE revert timeout 0
Interface-Type      Interface-Name      Oper-State Oper-Status
=====
Primary            1-7-19-0/eth        Standby     Trfc-Disable
Secondary          1-7-20-0/eth        Active      UP
```

- c View the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN      VLAN/SLAN  Bridge                                     St  Table Data
-----
dwn      Tagged 100      1-7-19-0-eth-100/bridge                  UP
1 bridges displayed
```

Notice that even though port 20 is now the active port, the name of the bridge does not change and displays the bridge as coming from port 19.

Create bridge interfaces on redundant Ethernet ports for intralink configurations

After creating the redundancy on the Ethernet ports, configure the primary Ethernet port with a bridge interface.

Configuring an intralink bridge interface on a redundant port

- 1 Create an intralink bridge interface on the primary port of the redundant pair.
 - a View the redundant pair.

```
zSH> line-red show 1-7-19-0/eth
redundancy status for 1-7-19-0/eth:
      NOREBOOT standbytx ENABLE timeout 0 NONREVERTIVE revert timeout
0
Interface-Type      Interface-Name      Oper-State Oper-Status
=====
Primary      1-7-19-0/eth      Active      UP
Secondary    1-7-20-0/eth      Standby     Trfc-Disable
```

- b Create the intralink bridge on the primary port.

```
zSH> bridge add 1-7-19-0/eth intralink vlan 100 tagged
Adding bridge on 1-7-19-0/eth
Created bridge-interface-record 1-7-19-0-eth-100/bridge
Bridge-path added successfully
```

Verify the bridge path.

```
zSH> bridge-path show
VLAN/SLAN  Bridge                                     Address
-----
      100 1-7-19-0-eth-100/bridge                  Intralink
```

- c Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN      VLAN/SLAN  Bridge                                     St  Table Data
```

```
-----
int          Tagged 100    1-7-19-0-eth-100/bridge      UP   S VLAN 100
Intralink
1 bridges displayed
```

2 Test line redundancy.

a Bounce the port.

```
zSH> port bounce 1-7-19-0/eth
1-7-19-0/eth set to admin state DOWN
1-7-19-0/eth set to admin state UP
```

b Show the line redundancy.

```
zSH> line-red show 1-7-19-0/eth
redundancy status for 1-7-19-0/eth:
      NOREBOOT standbytx ENABLE timeout 0 NONREVERTIVE revert timeout
0
Interface-Type      Interface-Name      Oper-State Oper-Status
=====
Primary             1-7-19-0/eth      Standby    Trfc-Disable
Secondary           1-7-20-0/eth      Active     UP
```

c View the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN      VLAN/SLAN  Bridge      St  Table Data
-----
int   Tagged 100      1-7-19-0-eth-100/bridge  UP  S VLAN 100
Intralink
1 bridges displayed
```

Notice that even though port 20 is now the active port, the name of the bridge does not change and displays the bridge as coming from port 19.

Create bridge interfaces on redundant Ethernet ports for TLS configurations

After creating the redundancy on the Ethernet ports, configure the primary Ethernet port with a bridge interface.

Configuring a TLS bridge interface on a redundant port

1 Create a TLS bridge interface on the primary port of the redundant pair.

a View the line redundancy.

```
zSH> line-red show 1-7-19-0/eth
redundancy status for 1-7-19-0/eth:
      NOREBOOT standbytx ENABLE timeout 0 NONREVERTIVE revert timeout
0
```

Interface-Type	Interface-Name	Oper-State	Oper-Status
=====	=====	=====	=====
Primary	1-7-19-0/eth	Active	UP
Secondary	1-7-20-0/eth	Standby	Trfc-Disable

b Create the TLS bridge on the primary port.

```
zSH> bridge add 1-7-19-0/eth tls vlan 100 tagged
Adding bridge on 1-7-19-0/eth
Created bridge-interface-record 1-7-19-0-eth-100/bridge
```

c Verify the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN      VLAN/SLAN  Bridge                               St  Table Data
-----
tls              Tagged 100   1-7-19-0-eth-100/bridge             UP
1 bridges displayed
```

2 Test the line redundancy.

a Bounce the active port.

```
zSH> port bounce 1-7-19-0/eth
1-7-19-0/eth set to admin state DOWN
1-7-19-0/eth set to admin state UP
```

b View the line redundancy.

```
zSH> line-red show 1-7-19-0/eth
redundancy status for 1-7-19-0/eth:
      NOREBOOT standbytx ENABLE timeout 0 NONREVERTIVE revert timeout
0
Interface-Type      Interface-Name      Oper-State  Oper-Status
=====
Primary             1-7-19-0/eth       Standby     Trfc-Disable
Secondary            1-7-20-0/eth       Active      UP
```

c View the bridge.

```
zSH> bridge show
Orig
Type VLAN/SLAN      VLAN/SLAN  Bridge                               St  Table Data
-----
tls              Tagged 100   1-7-19-0-eth-100/bridge             UP
1 bridges displayed
```

Notice that even though port 20 is now the active port, the name of the bridge does not change and displays the bridge as coming from port 19.

Removing redundant Ethernet ports

Removing a redundant Ethernet port

Redundancy may be removed from an Ethernet port, however there are limitations. The original primary port cannot be removed. Active ports can also not be removed.

To resolve downed ports which are on the primary port, resolve the problem with the port (whether downed link or card issue). Resolving the problem can include replacing the card with a new card, then running the **card add** command. When the new card comes up, the redundancy will be reestablished.

- 1 Show the current status of the redundancy group.

```
zSH> line-red show 1-7-19-0/eth
redundancy status for 1-7-19-0/eth:
      NOREBOOT standbytx ENABLE timeout 0 NONREVERTIVE revert timeout 0
Interface-Type      Interface-Name      Oper-State Oper-Status
=====
Primary            1-7-19-0/eth            Active     UP
Secondary          1-7-20-0/eth            Standby    Trfc-Disable
```

- 2 Remove the secondary port from the redundancy group.

```
zSH> line-red remove 1-7-20-0/eth
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
Interface 1-7-20-0/eth is no longer in a protection group.
```



Note: You cannot remove the primary port of a redundant pair even if it is in standby mode. You also cannot remove the secondary port when it is in the active mode.

- 3 Verify that the secondary port was removed from the redundant group.



Note: You should wait until you confirm that redundancy has been removed before changing any provisioning on the port. Verify the redundancy using one of the following show commands before adding or deleting bridge interfaces or IP interfaces on the Ethernet port.

```
zSH> line-red show 1-7-20-0/eth
redundancy status for 1-7-20-0/eth:
      NOREBOOT standbytx ENABLE timeout 0 NONREVERTIVE revert timeout 0
The 1-7-20-0/eth is not part of any redundancy group
```

Switchover from active to standby Ethernet port

A switchover from active to standby Ethernet port can be done automatically or forced manually using the **port bounce** command.

Automatically switched

A switchover is automatically triggered when a Loss of Signal occurs on the primary port.

When an automatic switchover occurs, an alarm is raised.

Manually switched

A switchover also can be manually forced with the **port bounce interface/type** command. This may occur to perform maintenance on the line.

```
zSH> port bounce 1-7-19-0/eth
1-7-19-0/eth set to admin state DOWN
1-7-19-0/eth set to admin state UP
```

A switchover can also be manually forced with the **port up interface/type** command.

```
zSH> port up 1-7-19-0/eth
1-7-19-0/eth set to admin state UP
```

Ethernet redundancy configuration limitations

The following limitations apply to Ethernet redundancy configurations.

- When a standby port is added, the configuration information is automatically inherited. If a port is configured as standby, logical interfaces cannot be configured on that port.
- The state of the card which has proposed secondary ports must be RUNNING. Before configuring a newly installed card, add the card with the **card add** command.
- A Ethernet port may only be a member of one redundancy group.
- An Ethernet port may only be made redundant with another Ethernet port.
- The following rules apply to deleting ports from Ethernet redundancy groups:
 - The primary port can not be deleted from the redundancy group.
 - An active port can not be deleted from the redundancy group. If the active port is the secondary port of the redundancy group, neither port can be removed.
 - Only the secondary port of a redundancy group can be deleted and only in the standby state.

- Upgrades cannot be scheduled on standby ports.



Note: If a switchover event is triggered when an upgrade is in progress, the upgrade is re-queued.

Port redundancy on Active Ethernet line cards

Ethernet facility protection now provides support for standby TX disable for Ethernet ports on line cards with optical SFPs only.

Laser on the SFP can be configured in receive only with transmit disabled. This facility protection is useful when the downstream device supports a 2x1 optical splitter and a single fiber terminates on the downstream device when two fibers are not needed at the end point.

Configuring a protection pair

Use the line-red command to create the protection pair on the Ethernet ports with optical SFPs.

1 Disable auto negotiate on the ports used for protection.

Auto negotiate must be disabled on the ports used for protection in order for redundancy to work.

```
zSH> update ether autonegstatus = disabled 1-5-1-0/eth
ether 1-5-1-0/eth
Record updated.
```

```
zSH> update ether autonegstatus = disabled 1-14-1-0/eth
ether 1-14-1-0/eth
Record updated.
```

2 Configure the protection pair on the Ethernet ports.

a Set the admin state to down on the port that will be the secondary port in the redundant protection pair.

```
zSH> port down 1-14-1-0/eth
1-14-1-0/eth set to admin state DOWN
```

b Configure the redundancy pair.

```
zSH> line-red add pri 1-5-1-0/eth sec 1-14-1-0/eth
This command may take several minutes to complete.
Do you want to continue? (yes or no) [no] yes
Protection pair has been created. 1-5-1-0/eth is primary and 1-14-1-0/eth is
secondary.
```

Views the line-red configuration.

```
zSH> line-red show 1-5-1-0/eth

redundancy status for 1-5-1-0/eth:
      NOREBOOT standbytx DISABLE timeout 0 NONREVERTIVE revert timeout 0
Interface-Type      Interface-Name      Oper-State Oper-Status
=====
Primary             1-5-1-0/eth             Active      UP
Secondary           1-14-1-0/eth            Down        Down
```


- c** Configure the primary port to *standbytx disabled*. This will turn off the transmit light until it lenses a link.

```
zSH> line-red set 1-5-1-0/eth standbytx disable
```

- d** Change the port status of the secondary port to up.

```
SH> port up 1-14-1-0/eth
1-14-1-0/eth set to admin state UP
```

- e** View the line-red configuration.

```
zSH> line-red show 1-5-1-0/eth
redundancy status for 1-5-1-0/eth:
      NOREBOOT standbytx DISABLE timeout 0 NONREVERTIVE revert timeout 0
Interface-Type      Interface-Name      Oper-State Oper-Status
=====
Primary            1-5-1-0/eth              Active     UP
Secondary          1-14-1-0/eth             Standby    Trfc-Disable
```

Default Ethernet alarms on line card Minor

The default alarm setting for Ethernet ports on Ethernet line cards has changed from *critical* to *minor*. The default alarm settings for Ethernet ports on uplink cards remains *critical*.

Enter the **port show alarm interface/type** command to view the level of alarm severity.

```
zSH> port show alarm 1-13-1-0/eth
-----
Interface              Alarm severity
-----
1-13-1-0/eth           MINOR
-----
```

Default alarm setting is *critical* for a uplink card Ethernet port.

```
zSH> port show alarm 1-a-5-0/eth
-----
Interface              Alarm severity
-----
1-a-5-0/eth            CRITICAL
-----
```

Settable alarm severity for Ethernet ports

The alarm severity for Ethernet ports can be set to the following levels: *critical*, *major*, *minor*, or *warning*.

Changing the alarm severity level for one Ethernet port

Use the **port config alarm interfaceName/type severity <severity level>** command to set the severity level on an Ethernet port.

- 1 View the current alarm setting on an Ethernet port.

```
zSH> port show alarm 1-13-1-0/eth
```

Interface	Alarm severity
1-13-1-0/eth	MINOR

- 2 Configure a different alarm setting on an Ethernet port.

```
zSH> port config alarm 1-13-1-0/eth severity major
Alarm severity level set for 1-13-1-0/eth is major
```

- 3 Verify the new alarm setting.

```
zSH> port show alarm 1-13-1-0/eth
```

Interface	Alarm severity
1-13-1-0/eth	MAJOR

Changing the alarm severity level for multiple Ethernet ports

Use the **port config alarm interfaceName/type severity <severity level>** command to set the severity level on multiple Ethernet ports.

- 1 View the alarm levels for all Ethernet ports on the uplink card.

```
zSH> port show alarm 1-a-**-*/eth
```

Interface	Alarm severity
1-a-1-0/eth	CRITICAL
1-a-11-0/eth	CRITICAL
1-a-10-0/eth	CRITICAL
1-a-9-0/eth	CRITICAL
1-a-8-0/eth	CRITICAL
1-a-7-0/eth	CRITICAL
1-a-6-0/eth	CRITICAL
1-a-5-0/eth	CRITICAL
1-a-4-0/eth	CRITICAL
1-a-3-0/eth	CRITICAL
1-a-2-0/eth	CRITICAL

- 2 View the alarm levels for all Ethernet ports on a line card.

```
zSH> port show alarm 1-6-**-*/eth
```

Interface	Alarm severity
1-6-1-0/eth	MINOR
1-6-20-0/eth	MINOR
1-6-19-0/eth	MINOR
1-6-18-0/eth	MINOR
1-6-17-0/eth	MINOR
1-6-16-0/eth	MINOR
1-6-15-0/eth	MINOR
1-6-14-0/eth	MINOR
1-6-13-0/eth	MINOR
1-6-12-0/eth	MINOR
1-6-11-0/eth	MINOR
1-6-10-0/eth	MINOR
1-6-9-0/eth	MINOR
1-6-8-0/eth	MINOR
1-6-7-0/eth	MINOR
1-6-6-0/eth	MINOR
1-6-5-0/eth	MINOR
1-6-4-0/eth	MINOR
1-6-3-0/eth	MINOR
1-6-2-0/eth	MINOR

3 Change the alarm setting of all Ethernet ports on the line card.

```

zSH> port config alarm 1-6-**-*/eth severity major
Alarm severity level set for 1-6-1-0/eth is major
Alarm severity level set for 1-6-20-0/eth is major
Alarm severity level set for 1-6-19-0/eth is major
Alarm severity level set for 1-6-18-0/eth is major
Alarm severity level set for 1-6-17-0/eth is major
Alarm severity level set for 1-6-16-0/eth is major
Alarm severity level set for 1-6-15-0/eth is major
Alarm severity level set for 1-6-14-0/eth is major
Alarm severity level set for 1-6-13-0/eth is major
Alarm severity level set for 1-6-12-0/eth is major
Alarm severity level set for 1-6-11-0/eth is major
Alarm severity level set for 1-6-10-0/eth is major
Alarm severity level set for 1-6-9-0/eth is major
Alarm severity level set for 1-6-8-0/eth is major
Alarm severity level set for 1-6-7-0/eth is major
Alarm severity level set for 1-6-6-0/eth is major
Alarm severity level set for 1-6-5-0/eth is major
Alarm severity level set for 1-6-4-0/eth is major
Alarm severity level set for 1-6-3-0/eth is major
Alarm severity level set for 1-6-2-0/eth is major

```

4 Verify the alarm severity level.

```

zSH> port show alarm 1-6-**-*/eth

```

Interface	Alarm severity

1-6-1-0/eth	MAJOR
1-6-20-0/eth	MAJOR
1-6-19-0/eth	MAJOR
1-6-18-0/eth	MAJOR
1-6-17-0/eth	MAJOR
1-6-16-0/eth	MAJOR
1-6-15-0/eth	MAJOR
1-6-14-0/eth	MAJOR
1-6-13-0/eth	MAJOR
1-6-12-0/eth	MAJOR
1-6-11-0/eth	MAJOR
1-6-10-0/eth	MAJOR
1-6-9-0/eth	MAJOR
1-6-8-0/eth	MAJOR
1-6-7-0/eth	MAJOR
1-6-6-0/eth	MAJOR
1-6-5-0/eth	MAJOR
1-6-4-0/eth	MAJOR
1-6-3-0/eth	MAJOR
1-6-2-0/eth	MAJOR

Enhanced Ethernet port statistics

Use **port stats** command to display or clear various statistical information.

port stats <ifName/Type> <intf|rmon|eth|all|clear>

The **port stats interface/type intf** command displays mib2 interface statistics.

See [Table 134 on page 1172](#) for parameter definitions.

zSH> port stats 1-1-19-0/eth intf	
Interface Name	1-1-19-0
Operational Status	Up
Received Bytes	660624000
Received Packets	436317
Received Multicast Packets	3830
Received Broadcast Packets	269
Transmitted Bytes	673299000
Transmitted Unicast Packets	448250
Transmitted Multicast Packets	307
Transmitted Broadcast Packets	309
Received Discards	1110
Received Errors	0
Received Unknown Protocols	0
Transmitted Discards	0
Transmitted Errors	0
Speed Bits per Second	*** n/a ***
Speed Megabits per Second	100

The **port stats interface/type rmon** command displays Ethernet remote monitoring statistics.

zSH> port stats 1-1-19-0/eth rmon

Total Dropped Events	0
Total Dropped Frames	0
Total Bytes	2115147000
Total Packets	1410098
Transmitted Packets	709274
Received Packets	700824
Transmitted Multicast Bytes	0
Received Multicast Bytes	5745000
Received Multicast Dropped Bytes	0
Transmitted Average Throughput	72672000
Received Average Throughput	72672000
Transmitted Bandwidth Occupancy	72
Received Bandwidth Occupancy	72
Total Broadcast Packets	578
Total Multicast Packets	4137
CRC Align Errors	0
Undersize Packets	0
Oversize Packets	0
Transmitted Oversize Packets	0
Received Oversize Packets	0
Fragments	0
Jabbers	0
Collisions	0
Transmitted No Errors	709274
Received No Errors	700824
IPMC Bridged Packets	3830
IPMC Routed Packets	0
Transmitted IPMC Dropped Packets	0
Received IPMC Dropped Packets	0
Total Packets 0 to 64 Bytes	0
Total Packets 65 to 127 Bytes	0
Total Packets 128 to 255 Bytes	0
Total Packets 256 to 511 Bytes	0
Total Packets 512 to 1023 Bytes	0
Total Packets 1024 to 1518 Bytes	1410098
Total Packets 1519 to 2047 Bytes	0
Total Packets 2048 to 4095 Bytes	0
Total Packets 4095 to 9216 Bytes	0
Received Packets 0 to 64 Bytes	0
Received Packets 65 to 127 Bytes	0
Received Packets 128 to 255 Bytes	0
Received Packets 256 to 511 Bytes	0
Received Packets 512 to 1023 Bytes	0
Received Packets 1024 to 1518 Bytes	700824
Received Packets 1519 to 2047 Bytes	0
Received Packets 2048 to 4095 Bytes	0
Received Packets 4095 to 9216 Bytes	0
Transmitted Packets 0 to 64 Bytes	0
Transmitted Packets 65 to 127 Bytes	0
Transmitted Packets 128 to 255 Bytes	0
Transmitted Packets 256 to 511 Bytes	0
Transmitted Packets 512 to 1023 Bytes	0
Transmitted Packets 1024 to 1518 Bytes	709274

Transmitted Packets 1519 to 2047 Bytes	0
Transmitted Packets 2048 to 4095 Bytes	0
Transmitted Packets 4095 to 9216 Bytes	0

The **port stats interface/type eth** command displays the Ethernet dot3 statistics.

```
zSH> port stats 1-1-19-0/eth eth
```

Alignment Errors	0
FCS Errors	0
Single Collision Frames	0
Multiple Collision Frames	0
SQE Test Errors	0
Deferred Transmissions	0
Late Collisions	0
Excessive Collisions	0
Internal Mac Transmit Errors	0
Carrier Sense Errors	0
FrameTooLongs	0
InternalMacReceiveErrors	0
SymbolErrors	0
DuplexStatus	Full

The **port stats interface/type all** commands displays all of the Ethernet statistics.

```
zSH> port stats 1-1-19-0/eth all
```

```
***** eth *****
```

Alignment Errors	0
FCS Errors	0
Single Collision Frames	0
Multiple Collision Frames	0
SQE Test Errors	0
Deferred Transmissions	0
Late Collisions	0
Excessive Collisions	0
Internal Mac Transmit Errors	0
Carrier Sense Errors	0
FrameTooLongs	0
InternalMacReceiveErrors	0
SymbolErrors	0
DuplexStatus	Full

```
***** rmon *****
```

Total Dropped Events	0
Total Dropped Frames	0
Total Bytes	3405022500
Total Packets	2270015
Transmitted Packets	1139233
Received Packets	1130782
Transmitted Multicast Bytes	0
Received Multicast Bytes	5745000
Received Multicast Dropped Bytes	0
Transmitted Average Throughput	71659832
Received Average Throughput	71659664

Transmitted Bandwidth Occupancy	71
Received Bandwidth Occupancy	71
Total Broadcast Packets	578
Total Multicast Packets	4137
CRC Align Errors	0
Undersize Packets	0
Oversize Packets	0
Transmitted Oversize Packets	0
Received Oversize Packets	0
Fragments	0
Jabbers	0
Collisions	0
Transmitted No Errors	1139233
Received No Errors	1130782
IPMC Bridged Packets	3830
IPMC Routed Packets	0
Transmitted IPMC Dropped Packets	0
Received IPMC Dropped Packets	0
Total Packets 0 to 64 Bytes	0
Total Packets 65 to 127 Bytes	0
Total Packets 128 to 255 Bytes	0
Total Packets 256 to 511 Bytes	0
Total Packets 512 to 1023 Bytes	0
Total Packets 1024 to 1518 Bytes	2270015
Total Packets 1519 to 2047 Bytes	0
Total Packets 2048 to 4095 Bytes	0
Total Packets 4095 to 9216 Bytes	0
Received Packets 0 to 64 Bytes	0
Received Packets 65 to 127 Bytes	0
Received Packets 128 to 255 Bytes	0
Received Packets 256 to 511 Bytes	0
Received Packets 512 to 1023 Bytes	0
Received Packets 1024 to 1518 Bytes	1130782
Received Packets 1519 to 2047 Bytes	0
Received Packets 2048 to 4095 Bytes	0
Received Packets 4095 to 9216 Bytes	0
Transmitted Packets 0 to 64 Bytes	0
Transmitted Packets 65 to 127 Bytes	0
Transmitted Packets 128 to 255 Bytes	0
Transmitted Packets 256 to 511 Bytes	0
Transmitted Packets 512 to 1023 Bytes	0
Transmitted Packets 1024 to 1518 Bytes	1139233
Transmitted Packets 1519 to 2047 Bytes	0
Transmitted Packets 2048 to 4095 Bytes	0
Transmitted Packets 4095 to 9216 Bytes	0
***** intf *****	
Interface Name	1-1-19-0
Operational Status	Up
Received Bytes	1696173000
Received Packets	1126682
Received Multicast Packets	3830
Received Broadcast Packets	269
Transmitted Bytes	1708849500
Transmitted Unicast Packets	1138617

Transmitted Multicast Packets	307
Transmitted Broadcast Packets	309
Received Discards	1110
Received Errors	0
Received Unknown Protocols	0
Transmitted Discards	0
Transmitted Errors	0
Speed Bits per Second	*** n/a ***
Speed Megabits per Second	100

The **port stats clear** *interface/type* command clears all port stats counters.

```
zSH> port stats clear 1-1-19-0/eth
INTF Stats cleared
```

[Table 134](#) defines the parameters for all of the Ethernet statistics.

Table 134: MXK Enhanced Ethernet port statistics

Parameter	Description
eth	
Alignment Errors	A count of frames received on a particular interface that are not an integral number of octets in length and do not pass the FCS check. The count represented by an instance of this object is incremented when the alignment Error status is returned by the MAC service to the LLC (or other MAC user). Received frames for which multiple error conditions obtain are, according to the conventions of IEEE 802.3 Layer Management, counted exclusively according to the error status presented to the LLC. This counter does not increment for 8-bit wide group encoding schemes. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of <code>ifCounterDiscontinuityTime</code>.
FCS Errors	A count of frames received on a particular interface that are an integral number of octets in length but do not pass the FCS check. This count does not include frames received with frame-too-long or frame-too-short error. The count represented by an instance of this object is incremented when the <code>frameCheckError</code> status is returned by the MAC service to the LLC (or other MAC user). Received frames for which multiple error conditions obtain are, according to the conventions of IEEE 802.3 Layer Management, counted exclusively according to the error status presented to the LLC. Note: Coding errors detected by the physical layer for speeds above 10 Mb/s will cause the frame to fail the FCS check. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of <code>ifCounterDiscontinuityTime</code>.

Table 134: MXK Enhanced Ethernet port statistics (Continued)

Parameter	Description
Single Collision Frames	A count of successfully transmitted frames on a particular interface for which transmission is inhibited by exactly one collision. A frame that is counted by an instance of this object is also counted by the corresponding instance of either the ifOutUcastPkts, ifOutMulticastPkts, or ifOutBroadcastPkts, and is not counted by the corresponding instance of the dot3StatsMultipleCollisionFrames object. This counter does not increment when the interface is operating in full-duplex mode. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Multiple Collision Frames	A count of successfully transmitted frames on a particular interface for which transmission is inhibited by more than one collision. A frame that is counted by an instance of this object is also counted by the corresponding instance of either the ifOutUcastPkts, ifOutMulticastPkts, or ifOutBroadcastPkts, and is not counted by the corresponding instance of the dot3StatsSingleCollisionFrames object. This counter does not increment when the interface is operating in full-duplex mode. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
SQE Test Errors	A count of times that the SQE TEST ERROR message is generated by the PLS sublayer for a particular interface. The SQE TEST ERROR is set in accordance with the rules for verification of the SQE detection mechanism in the PLS Carrier Sense Function as described in IEEE Std. 802.3, 1998 Edition, section 7.2.4.6. This counter does not increment on interfaces operating at speeds greater than 10 Mb/s, or on interfaces operating in full-duplex mode. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Deferred Transmissions	A count of frames for which the first transmission attempt on a particular interface is delayed because the medium is busy. The count represented by an instance of this object does not include frames involved in collisions. This counter does not increment when the interface is operating in full-duplex mode. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Late Collisions	The number of times that a collision is detected on a particular interface later than one slotTime into the transmission of a packet. A (late) collision included in a count represented by an instance of this object is also considered as a (generic) collision for purposes of other collision-related statistics. This counter does not increment when the interface is operating in full-duplex mode. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

Table 134: MXK Enhanced Ethernet port statistics (Continued)

Parameter	Description
Excessive Collisions	A count of frames for which transmission on a particular interface fails due to excessive collisions. This counter does not increment when the interface is operating in full-duplex mode. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Internal Mac Transmit Errors	A count of frames for which transmission on a particular interface fails due to an internal MAC sublayer transmit error. A frame is only counted by an instance of this object if it is not counted by the corresponding instance of either the dot3StatsLateCollisions object, the dot3StatsExcessiveCollisions object, or the dot3StatsCarrierSenseErrors object. The precise meaning of the count represented by an instance of this object is implementation- specific. In particular, an instance of this object may represent a count of transmission errors on a particular interface that are not otherwise counted. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Carrier Sense Errors	The number of times that the carrier sense condition was lost or never asserted when attempting to transmit a frame on a particular interface. The count represented by an instance of this object is incremented at most once per transmission attempt, even if the carrier sense condition fluctuates during a transmission attempt. This counter does not increment when the interface is operating in full-duplex mode. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
FrameTooLongs	A count of frames received on a particular interface that exceed the maximum permitted frame size. The count represented by an instance of this object is incremented when the frameTooLong status is returned by the MAC service to the LLC (or other MAC user). Received frames for which multiple error conditions obtain are, according to the conventions of IEEE 802.3 Layer Management, counted exclusively according to the error status presented to the LLC. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

Table 134: MXK Enhanced Ethernet port statistics (Continued)

Parameter	Description
InternalMacReceiveErrors	A count of frames for which reception on a particular interface fails due to an internal MAC sublayer receive error. A frame is only counted by an instance of this object if it is not counted by the corresponding instance of either the dot3StatsFrameTooLongs object, the dot3StatsAlignmentErrors object, or the dot3StatsFCSErrors object. The precise meaning of the count represented by an instance of this object is implementation- specific. In particular, an instance of this object may represent a count of receive errors on a particular interface that are not otherwise counted. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime
SymbolErrors	For an interface operating at 100 Mb/s, the number of times there was an invalid data symbol when a valid carrier was present. For an interface operating in half-duplex mode at 1000 Mb/s, the number of times the receiving media is non-idle (a carrier event) for a period of time equal to or greater than slotTime, and during which there was at least one occurrence of an event that causes the PHY to indicate 'Data reception error' or 'carrier extend error' on the GMII. For an interface operating in full-duplex mode at 1000 Mb/s, the number of times the receiving media is non-idle a carrier event) for a period of time equal to or greater than minFrameSize, and during which there was at least one occurrence of an event that causes the PHY to indicate 'Data reception error' on the GMII. The count represented by an instance of this object is incremented at most once per carrier event, even if multiple symbol errors occur during the carrier event. This count does not increment if a collision is present. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
DuplexStatus	The current mode of operation of the MAC entity. 'unknown' indicates that the current duplex mode could not be determined. Management control of the duplex mode is accomplished through the MAU MIB. When an interface does not support autonegotiation, or when autonegotiation is not enabled, the duplex mode is controlled using ifMauDefaultType. When autonegotiation is supported and enabled, duplex mode is controlled using ifMauAutoNegAdvertisedBits. In either case, the currently operating duplex mode is reflected both in this object and in ifMauType. Note that this object provides redundant information with ifMauType. Normally, redundant objects are discouraged. However, in this instance, it allows a management application to determine the duplex status of an interface without having to know every possible value of ifMauType. This was felt to be sufficiently valuable to justify the redundancy. Values: unknown halfDuplex fullDuplex
rmon	Remote Network Monitoring

Table 134: MXK Enhanced Ethernet port statistics (Continued)

Parameter	Description
Total Dropped Events	The total number of events in which packets were dropped by the probe due to lack of resources. Note that this number is not necessarily the number of packets dropped; it is just the number of times this condition has been detected.
Total Dropped Frames	The total number of frames that were received by the probe and therefore not accounted for in the <code>zhoneEtherStatsDropEvents</code>, but that the probe chose not to count for this entry for whatever reason. Most often, this event occurs when the probe is out of some resources and decides to shed load from this collection. This count does not include packets that were not counted because they had MAC-layer errors. Note that, unlike the <code>dropEvents</code> counter, this number is the exact number of frames dropped.
Total Bytes	The total number of octets of data (including those in bad packets) transmitted and received on the network (excluding framing bits but including FCS octets). This object can be used as a reasonable estimate of 10-Megabit ethernet utilization. If greater precision is desired, the <code>zhoneEtherStatsPkts</code> and <code>zhoneEtherStatsOctets</code> objects should be sampled before and after a common interval. The differences in the sampled values are <code>Pkts</code> and <code>Octets</code>, respectively, and the number of seconds in the interval is <code>Interval</code>. These values are used to calculate the Utilization as follows: $\text{Pkts} * (9.6 + 6.4) + (\text{Octets} * .8)$ $\text{Utilization} = \frac{\text{Interval} * 10,000}{\text{Interval} * 10,000}$ The result of this equation is the value <code>Utilization</code> which is the percent utilization of the ethernet segment on a scale of 0 to 100 percent.
Total Packets	The total number of packets (including bad packets, broadcast packets, and multicast packets) transmitted and received.
Transmitted Packets	The total number of packets (including bad packets, broadcast packets, and multicast packets) transmitted.
Received Packets	The total number of packets (including bad packets, broadcast packets, and multicast packets) received.
Transmitted Multicast Bytes	Transmitted multicast bytes.
Received Multicast Bytes	Received multicast bytes.
Received Multicast Dropped Bytes	Dropped multicast bytes.
Transmitted Average Throughput	Average transmit throughput in bits per second since last query. For accuracy purposes, it is recommended that this object be queried in intervals of five (5) seconds or greater.

Table 134: MXK Enhanced Ethernet port statistics (Continued)

Parameter	Description
Received Average Throughput	Average receive throughput in bits per second since last query. For accuracy purposes, it is recommended that this object be queried in intervals of five (5) seconds or greater.
Transmitted Bandwidth Occupancy	Percentage of bandwidth currently being utilized for transmitting traffic. This rate is calculated based on the delta between prior and current query of this object. For accuracy purposes, it is recommended that this object be queried in intervals of five (5) seconds or greater.
Received Bandwidth Occupancy	Percentage of bandwidth currently being utilized for receiving traffic. This rate is calculated based on the delta between prior and current query of this object. For accuracy purposes, it is recommended that this object be queried in intervals of five (5) seconds or greater.
Total Broadcast Packets	The total number of good packets transmitted and received that were directed to the broadcast address. Note that this does not include multicast packets.
Total Multicast Packets	The total number of good packets transmitted and received that were directed to a multicast address. Note that this number does not include packets directed to the broadcast address.
CRC Align Errors	The total number of packets received that had a length (excluding framing bits, but including FCS octets) of between 64 and 1518 octets, inclusive, but had either a bad Frame Check Sequence (FCS) with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error).
Undersize Packets	The total number of packets received that were less than 64 octets long (excluding framing bits, but including FCS octets) and were otherwise well formed.
Oversize Packets	The total number of packets transmitted and received that were longer than 1518 octets (excluding framing bits, but including FCS octets) and were otherwise well formed.
Transmitted Oversize Packets	The total number of packets transmitted that were longer than 1518 octets (excluding framing bits, but including FCS octets) and were otherwise well formed.
Received Oversize Packets	The total number of packets received that were longer than 1518 octets (excluding framing bits, but including FCS octets) and were otherwise well formed.
Fragments	The total number of packets received that were less than 64 octets in length (excluding framing bits but including FCS octets) and had either a bad Frame Check Sequence (FCS) with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). Note that it is entirely normal for <code>zhoneEtherStatsFragments</code> to increment. This is because it counts both runts (which are normal occurrences due to collisions) and noise hits.

Table 134: MXK Enhanced Ethernet port statistics (Continued)

Parameter	Description
Jabbers	The total number of packets received that were longer than 1518 octets (excluding framing bits, but including FCS octets), and had either a bad Frame Check Sequence (FCS) with an integral number of octets (FCS Error) or a bad FCS with a non-integral number of octets (Alignment Error). Note that this definition of jabber is different than the definition in IEEE-802.3 section 8.2.1.5 (10BASE5) and section 10.3.1.4 (10BASE2). These documents define jabber as the condition where any packet exceeds 20 ms. The allowed range to detect jabber is between 20 ms and 150 ms.
Collisions	The best estimate of the total number of collisions on this Ethernet segment. The value returned will depend on the location of the RMON probe. Section 8.2.1.3 (10BASE-5) and section 10.3.1.3 (10BASE-2) of IEEE standard 802.3 states that a station must detect a collision, in the receive mode, if three or more stations are transmitting simultaneously. A repeater port must detect a collision when two or more stations are transmitting simultaneously. Thus a probe placed on a repeater port could record more collisions than a probe connected to a station on the same segment would. Probe location plays a much smaller role when considering 10BASE-T. 14.2.1.4 (10BASE-T) of IEEE standard 802.3 defines a collision as the simultaneous presence of signals on the DO and RD circuits (transmitting and receiving at the same time). A 10BASE-T station can only detect collisions when it is transmitting. Thus probes placed on a station and a repeater, should report the same number of collisions. Note also that an RMON probe inside a repeater should ideally report collisions between the repeater and one or more other hosts (transmit collisions as defined by IEEE 802.3k) plus receiver collisions observed on any coax segments to which the repeater is connected.
Transmitted No Errors	The total number of TX packets transmitted without error.
Received No Errors	The total number of RX packets received without error.
IPMC Bridged Packets	Broadcom IPMC Bridged Packet count.
IPMC Routed Packets	Broadcom IPMC Routed Packet count.
Transmitted IPMC Dropped Packets	Broadcom IPMC Tx Dropped Packet count.
Received IPMC Dropped Packets	Broadcom IPMC Rx Dropped Packet count.
Total Packets 0 to 64 Bytes	The total number of packets (including bad packets) transmitted and received that were 64 octets in length (excluding framing bits but including FCS octets).
Total Packets 65 to 127 Bytes	The total number of packets (including bad packets) transmitted and received that were between 65 and 127 octets in length inclusive (excluding framing bits but including FCS octets).
Total Packets 128 to 255 Bytes	The total number of packets (including bad packets) transmitted and received that were between 128 and 255 octets in length inclusive (excluding framing bits but including FCS octets).

Table 134: MXK Enhanced Ethernet port statistics (Continued)

Parameter	Description
Total Packets 256 to 511 Bytes	The total number of packets (including bad packets) transmitted and received that were between 256 and 511 octets in length inclusive (excluding framing bits but including FCS octets).
Total Packets 512 to 1023 Bytes	The total number of packets (including bad packets) transmitted and received that were between 512 and 1023 octets in length inclusive (excluding framing bits but including FCS octets).
Total Packets 1024 to 1518 Bytes	The total number of packets (including bad packets) transmitted and received that were between 1024 and 1518 octets in length inclusive (excluding framing bits but including FCS octets).
Total Packets 1519 to 2047 Bytes	The total number of packets (including bad packets) transmitted and received that were between 1519 and 2047 octets in length inclusive (excluding framing bits but including FCS octets).
Total Packets 2048 to 4095 Bytes	The total number of packets (including bad packets) transmitted and received that were between 2048 and 4095 octets in length inclusive (excluding framing bits but including FCS octets).
Total Packets 4096 to 9216 Bytes	The total number of packets (including bad packets) transmitted and received that were between 4096 and 9216 octets in length inclusive (excluding framing bits but including FCS octets).
Received Packets 0 to 64 Bytes	The total number of packets (including bad packets) received that were between 0 and 64 octets in length inclusive (excluding framing bits but including FCS octets).
Received Packets 65 to 127 Bytes	The total number of packets (including bad packets) received that were between 65 and 127 octets in length inclusive (excluding framing bits but including FCS octets).
Received Packets 128 to 255 Bytes	The total number of packets (including bad packets) received that were between 128 and 255 octets in length inclusive (excluding framing bits but including FCS octets).
Received Packets 256 to 511 Bytes	The total number of packets (including bad packets) received that were between 256 and 511 octets in length inclusive (excluding framing bits but including FCS octets).
Received Packets 512 to 1023 Bytes	The total number of packets (including bad packets) received that were between 512 and 1023 octets in length inclusive (excluding framing bits but including FCS octets).
Received Packets 1024 to 1518 Bytes	The total number of packets (including bad packets) received that were between 1024 and 1518 octets in length inclusive (excluding framing bits but including FCS octets).
Received Packets 1519 to 2047 Bytes	The total number of packets (including bad packets) received that were between 1519 and 2047 octets in length inclusive (excluding framing bits but including FCS octets).
Received Packets 2048 to 4095 Bytes	The total number of packets (including bad packets) received that were between 2048 and 4095 octets in length inclusive (excluding framing bits but including FCS octets).
Received Packets 4096 to 9216 Bytes	The total number of packets (including bad packets) received that were between 4096 and 9216 octets in length inclusive (excluding framing bits but including FCS octets).
Transmitted Packets 0 to 64 Bytes	The total number of packets (including bad packets) transmitted that were between 0 and 64 octets in length inclusive (excluding framing bits but including FCS octets).

Table 134: MXK Enhanced Ethernet port statistics (Continued)

Parameter	Description
Transmitted Packets 65 to 127 Bytes	The total number of packets (including bad packets) transmitted that were between 65 and 127 octets in length inclusive (excluding framing bits but including FCS octets).
Transmitted Packets 128 to 255 Bytes	The total number of packets (including bad packets) transmitted that were between 128 and 255 octets in length inclusive (excluding framing bits but including FCS octets).
Transmitted Packets 256 to 511 Bytes	The total number of packets (including bad packets) transmitted that were between 256 and 511 octets in length inclusive (excluding framing bits but including FCS octets).
Transmitted Packets 512 to 1023 Bytes	The total number of packets (including bad packets) transmitted that were between 512 and 1023 octets in length inclusive (excluding framing bits but including FCS octets).
Transmitted Packets 1024 to 1518 Bytes	The total number of packets (including bad packets) transmitted that were between 1024 and 1518 octets in length inclusive (excluding framing bits but including FCS octets).
Transmitted Packets 1519 to 2047 Bytes	The total number of packets (including bad packets) transmitted that were between 1519 and 2047 octets in length inclusive (excluding framing bits but including FCS octets).
Transmitted Packets 2048 to 4095 Bytes	The total number of packets (including bad packets) transmitted that were between 2048 and 4095 octets in length inclusive (excluding framing bits but including FCS octets).
Transmitted Packets 4096 to 9216 Bytes	The total number of packets (including bad packets) transmitted that were between 4096 and 9216 octets in length inclusive (excluding framing bits but including FCS octets).
intf	Interface statistics
Interface Name	The textual name of the interface. The value of this object should be the name of the interface as assigned by the local device and should be suitable for use in commands entered at the device's `console`. This might be a text name, such as `le0` or a simple port number, such as `1`, depending on the interface naming syntax of the device. If several entries in the ifTable together represent a single interface as named by the device, then each will have the same value of ifName. Note that for an agent which responds to SNMP queries concerning an interface on some other (proxied) device, then the value of ifName for such an interface is the proxied device's local name for it. If there is no local name, or this object is otherwise not applicable, then this object contains a zero-length string.

Table 134: MXK Enhanced Ethernet port statistics (Continued)

Parameter	Description
Operational Status	The current operational state of the interface. The testing(3) state indicates that no operational packets can be passed. If ifAdminStatus is down(2) then ifOperStatus should be down(2). If ifAdminStatus is changed to up(1) then ifOperStatus should change to up(1) if the interface is ready to transmit and receive network traffic; it should change to dormant(5) if the interface is waiting for external actions (such as a serial line waiting for an incoming connection); it should remain in the down(2) state if and only if there is a fault that prevents it from going to the up(1) state; it should remain in the notPresent(6) state if the interface has missing (typically, hardware) components. Values: up down testing unknown dormant notPresent lowerLayerDown
Received Bytes	The total number of octets received on the interface, including framing characters. This object is a 64-bit version of ifInOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Received Multicast Packets	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a multicast address at this sub-layer. For a MAC layer protocol, this includes both Group and Functional addresses. This object is a 64-bit version of ifInMulticastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Received Broadcast Packets	The number of packets, delivered by this sub-layer to a higher (sub-)layer, which were addressed to a broadcast address at this sub-layer. This object is a 64-bit version of ifInBroadcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Transmitted Bytes	The total number of octets transmitted out of the interface, including framing characters. This object is a 64-bit version of ifOutOctets. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

Table 134: MXK Enhanced Ethernet port statistics (Continued)

Parameter	Description
Transmitted Unicast Packets	The total number of packets that higher-level protocols requested be transmitted, and which were not addressed to a multicast or broadcast address at this sub-layer, including those that were discarded or not sent. This object is a 64-bit version of ifOutUcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Transmitted Multicast Packets	The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a multicast address at this sub-layer, including those that were discarded or not sent. For a MAC layer protocol, this includes both Group and Functional addresses. This object is a 64-bit version of ifOutBroadcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Transmitted Broadcast Packets	The total number of packets that higher-level protocols requested be transmitted, and which were addressed to a broadcast address at this sub-layer, including those that were discarded or not sent. This object is a 64-bit version of ifOutBroadcastPkts. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Received Discards	The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Received Errors	For packet-oriented interfaces, the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. For character-oriented or fixed-length interfaces, the number of inbound transmission units that contained errors preventing them from being deliverable to a higher-layer protocol. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Received Unknown Protocols	For packet-oriented interfaces, the number of packets received via the interface which were discarded because of an unknown or unsupported protocol. For character-oriented or fixed-length interfaces that support protocol multiplexing the number of transmission units received via the interface which were discarded because of an unknown or unsupported protocol. For any interface that does not support protocol multiplexing, this counter will always be 0. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.

Table 134: MXK Enhanced Ethernet port statistics (Continued)

Parameter	Description
Transmitted Discards	The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Transmitted Errors	For packet-oriented interfaces, the number of outbound packets that could not be transmitted because of errors. For character-oriented or fixed-length interfaces, the number of outbound transmission units that could not be transmitted because of errors. Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the value of ifCounterDiscontinuityTime.
Speed Bits per Second	An estimate of the interface's current bandwidth in bits per second. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. If the bandwidth of the interface is greater than the maximum value reportable by this object then this object should report its maximum value (4,294,967,295) and ifHighSpeed must be used to report the interface's speed. For a sub-layer which has no concept of bandwidth, this object should be zero.
Speed Megabits per Second	An estimate of the interface's current bandwidth in units of 1,000,000 bits per second. If this object reports a value of `n' then the speed of the interface is somewhere in the range of `n-500,000' to `n+499,999'. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. For a sub-layer which has no concept of bandwidth, this object should be zero.

13

MXK ADSL2+ BOND CARDS

This chapter describes the MXK ADSL2+ bond cards and ADSL card configuration:

- [ADSL2+ bond cards, page 1185](#)
- [ADSL2+ on the MXK, page 1202](#)
- [ADSL2+ interface configuration, page 1213](#)
- [ADSL2+ 48-port bonding, page 1272](#)
- [ADSL2+ 72-port bonding, page 1276](#)
- [ADSL2+ POTS line card ATM, page 1279](#)
- [ADSL2+ statistics, page 1283](#)
- [ADSL2+ Cabinet Mode, page 1295](#)
- [Downstream Power Backoff \(DPBO\), page 1299](#)
- [ADSL2+ cable and port pinouts, page 1299](#)
- [ADSL2+ testing \(SELT/DELT\) on the MXK, page 1334](#)

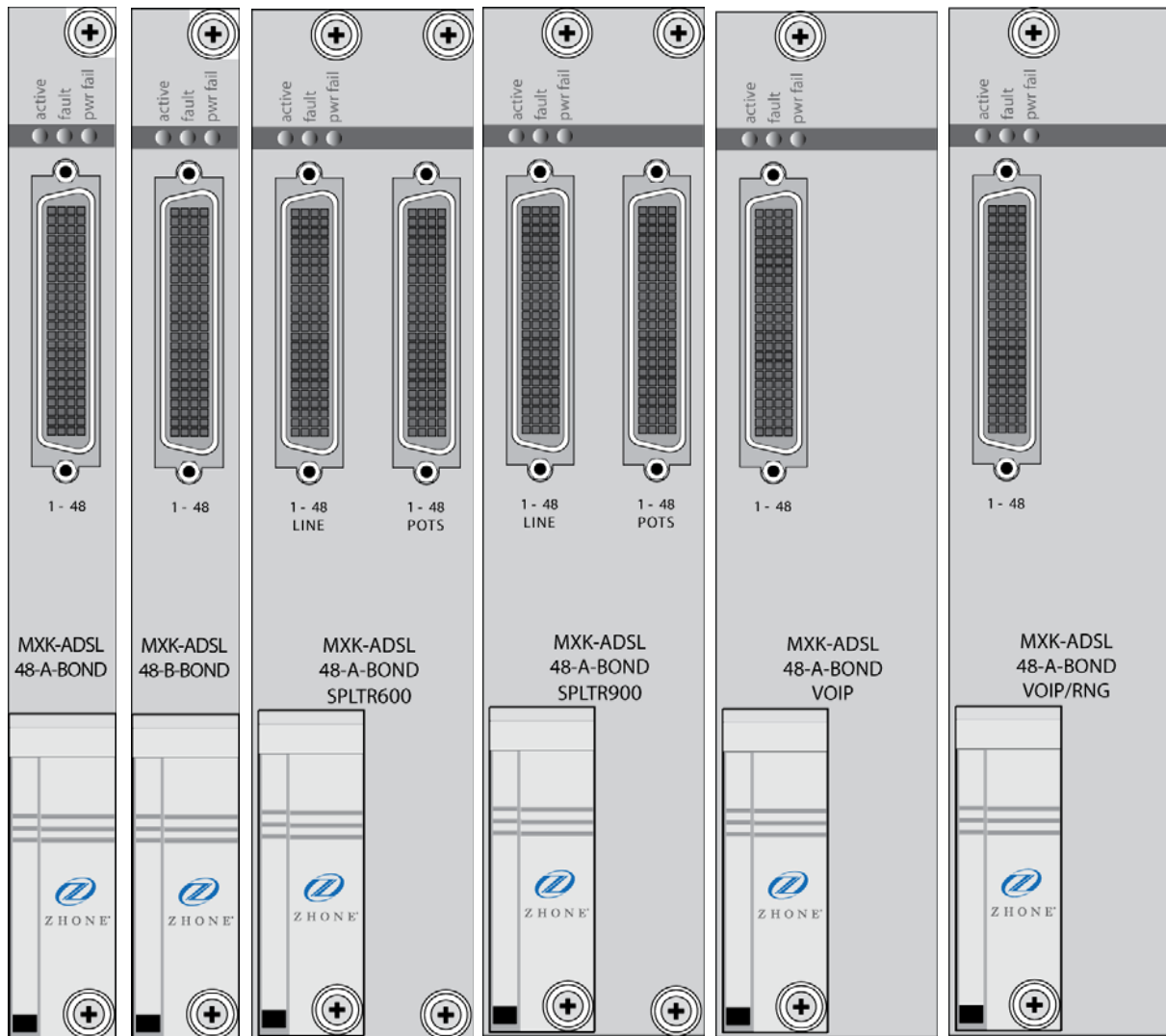
ADSL2+ bond cards

This section describes the MXK ADSL2+ bond cards with splitters and without splitters and how to configure the ADSL interfaces.

Configuration of the ADSL combo cards providing POTS and VoIP is discussed in *Chapter 14, MXK POTS Cards*.

- [ADSL2+ bond 48-port card overview, page 1186](#)
- [ADSL2+ bond 48-port card specifications, page 1187](#)
- [ADSL2+ bond 48-port card configuration, page 1193](#)
- [View additional card information, page 1195](#)

ADSL2+ bond 48-port card overview



The ADSL2+ bond 48-port cards are:

- MXK-ADSL2+-BCM-48A
- MXK-ADSL2+-BCM-48B
- MXK-ADSL2+-SPLTR600-BCM-48A-2S
- MXK-ADSL2+-SPLTR900-BCM-48A-2S

MXK-ADSL2+-BCM-48A

MXK-ADSL2+-BCM-48B

These cards are a single slot card that supports ADSL2+ Annex A/M or ADSL2+ Annex B. The standards supported are ANSI T1.413 Issue 2, G.992.1 (G.dmt), G.992.2 (G.lite), and ADSL2+ (G.992.5) standards.

MXK-ADSL2+-POTS-BCM-48A-2S**MXK-ADSL2+-POTS-BCM-48A-RNG-2S**

These are two-slot cards and to provide 48 ports of integrated ADSL and POTS VoIP services. This card supports the ANSI T1.413 Issue 2, G.992.1 (G.dmt) and G.992.2 (G.lite), G.992.3 and G.992.4 (ADSL2), G.992.5 (ADSL2+), Annex A, Annex B, and Annex M ADSL standards. Also supports SIP, SIP-PLAR, H.248, MGCP protocols and H.248 (MEGACO) protocols.

The MXK-ADSL2+-POTS-BCM-48A-RNG-2S also provides integrated ringing functionality and the internal line testing functionality (same capabilities as the enhanced MTAC/ TAC ITM card). Integrated ringing functionality on the line card means that the total number of POTS lines on an MXK chassis that can be in the ringing state simultaneously is increased as well as simplifying the effort required to match the ringing capacity of the MTAC cards to the POTS lines on the shelf. Also, when the POTS lines in the chassis are provisioned on this card, a separate TAC/RING card is not needed in the chassis, which increases the line capacity of the shelf and reduces the per port costs of deployment.

MXK-ADSL2+-SPLTR600-BCM-48-2S**MXK-ADSL2+-SPLTR900-BCM-48-2S**

These cards are two-slot cards with an integrated POTS splitter to provide 48 ports of integrated ADSL and POTS service. Each of these lines are combined with the ADSL2+ signal internally and exits the line card in the subscriber direction with both ADSL and POTS on the loop. In the network direction POTS is split from the ADSL signal keeping POTS on copper pairs and placing the ADSL data information on the IP network.

The MXK-ADSL2+-SPLTR600-BCM-48-2S, and MXK-ADSL2+-SPLTR900-BCM-48-2S cards support the ANSI T1.413 Issue 2, G.992.1 (G.dmt) and G.992.2 (G.lite), G.992.3 and G.992.4 (ADSL2), G.992.5 (ADSL2+), Annex A standards and Annex M ADSL standards.

The MXK-ADSL2+-BCM-48A, MXK-ADSL2+-SPLTR600-BCM-48-2S, and MXK-ADSL2+-SPLTR900-BCM-48-2S cards support VoIP POTS services.



Caution: Use caution when inserting and removing the ADSL2+ cards to and from the MXK chassis. Be careful not to damage the caps on the board.

ADSL2+ bond 48-port card specifications

Table 135 provides the specifications for the MXK-ADSL2+-BCM-48A, MXK-ADSL2+-BCM-48B, MXK-ADSL2+-SPLTR600-BCM-48-2S, MXK-ADSL2+-SPLTR900-BCM-48-2S, MXK-ADSL2+-POTS-BCM-48A-2S, MXK-ADSL2+-POTS-BCM-48A-RNG-2S, and MXK-ADSL2+-POTS-BCM-48B-2S.

Table 135: ADSL-48-BCM cards specifications

Specification	Description
Size	1 slot: MXK-ADSL-BCM-48A 2 slot: MXK-ADSL2+-SPLTR600-BCM-48-2S, MXK-ADSL2+-SPLTR900-BCM-48-2S, MXK-ADSL2+-POTS-BCM-48A-2S, and MXK-ADSL2+-POTS-BCM-48B-2S MXK-ADSL2+-POTS-BCM-48A-RNG-2S
Density	48 ports ADSL
Connectors	1 slot: One 96-pin telco connector 2 slot: Two 96-pin telco connectors
Standards supported	ANSI T1.413.2 G.992.1 (G.DMT) G.992.2 (G.Lite) G.992.3 and G.992.4 (ADSL2) G.992.5 (ADSL2+)
Line characteristics	Annex A supported (ADSL2+ over POTS) Annex B supported (ADSL2+ over ISDN) Annex A/M supported (ADSL2/ADSL2+) Fast Path or Interleaved mode supported on a per port basis Fast Retrain supported

Table 135: ADSL-48-BCM cards specifications (Continued)

Specification	Description
Supported train rates	T1.413 (fast and interleaved): - up to 8192 Kbps downstream - up to 1024 Kbps upstream G.lite (interleaved only) - up to 1536Kbps downstream - up to 512Kbps upstream G.DMT (interleave): - up to 12288Kbps downstream - up to 1331.2Kbps upstream G.DMT (fast): - up to 80192Kbps downstream - up to 1024Kbps upstream ADSL2: - Annex A: up to 12288 Kbps downstream - Annex A: up to 1331.2 Kbps upstream - Annex B: 32 Kbps to 28 Mbps downstream 32 Kbps to 1.2 Mbps upstream ADSL2+: - Annex A: up to 28672Kbps downstream - Annex A: up to 1024Kbps upstream - Annex B: 32 Kbps to 28 Mbps downstream - Annex B: 32 Kbps to 1.2 Mbps upstream - Annex M: up to 24576Kbps downstream - Annex M: up to 3584Kbps upstream In the case of Annex M, the upstream will vary depending on how many bins are allocated from the downstream to the upstream. Typically the best case for increasing the upstream bandwidth is approximately 2252.8Kbps.
Metallic test function	Look-out test
Power ADSL2+	23 W nominal 38.16 W maximum total. This is at maximum distance with all ports trained at ADSL2+ rates
Power ADSL2+ splitter	25.92 W nominal 57.40 W maximum total. This is at maximum distance with all ports trained at ADSL2+ rates

Table 135: ADSL-48-BCM cards specifications (Continued)

Specification	Description
Power ADSL2+ POTS	54 W nominal 143.28 W maximum total. This is at maximum distance with all ports trained at ADSL2+ rates
Chip set	Broadcom

ADSL+POTS combo card configuration

Each card installed in the system must have a **card-profile**. The line card type determines the parameter settings in the **card-profile** and the software image for the card. Performing a **card add** <slot #> automatically creates the **card-profile** for the card with the correct software image and settings.

[Table 137](#) shows the card type and software image for the ADSL2+-POTS combo cards on the MXK.

Table 136: MXK ADSL POTS combo card types

Card	Type	Name of software image
MXK-ADSL2+-POTS-BCM-48A-2S	10202	<i>mxlc48aadslbond.bin</i>
MXK-ADSL2+-POTS-BCM-48A-RNG-2S	10202	<i>mxlc48badslbond.bin</i>

The card line types of 48-port ADSL2+ POTS combo cards on the MXK are:

- **unknowntype** (default)
- **adsl-pots-pv** (ADSL with VoIP)
- **adsl-pots-pv-rng-itm** (ADSL with VoIP, and integrated ringing generation and line testing. For MXK-ADSL2+-POTS-BCM-48A-RNG-2S card only)

For MXK-ADSL2+-POTS-BCM-48A-RNG-2S card:

- Both **adsl-pots-pv** or **adsl-pots-pv-rng-itm** will always use the internal ring generator on the card.
- By provisioning a card line type parameter to **adsl-pots-pv** for the MXK-ADSL2+-POTS-BCM-48A-RNG-2S card, it will cause this RNG combo card to behave exactly as the non-RNG versions of ADSL POTS combo cars from a loop test perspective. In this mode therefore, loop testing can be achieved through external test heads (like Tollgrade) from test access ports on the MTAC/TAC cards. Alternatively, you can use the integrated Test Module (ITM) functionality on the MTAC/TAC cards to perform look out testing on the RNG combo cards.

- By provisioning a card line type parameter to **adsl-pots-pv-rng-itm** for the RNG combo card, it will cause the RNG combo card use the integrated ITM test functionality on the RNG card. In this mode, access to the test functionality of the MTAC/TAC cards (either the ITM or the external test access ports) is blocked.

Adding ADSL-POTS-RNG-Combo cards

The following example adds an MXK-ADSL2+-POTS-BCM-48A-RNG-2S card to the system:

- 1 Install the MXK-ADSL2+-POTS-BCM-48A-RNG-2S card in the desired line card slot.
- 2 Create a **card-profile** for the card:

```
zSH> new card-profile 10
card-profile 1/10/10202
Please provide the following: [q]uit.
sw-file-name: -----> {mxlc48aadslbond.bin}:
admin-status: -----> {operational}:
upgrade-sw-file-name: ---> {}:
upgrade-vers: -----> {}:
admin-status-enable: ----> {enable}:
sw-upgrade-admin: -----> {reloadcurrrev}:
sw-enable: -----> {true}:
sw-upgrade-enable: -----> {false}:
card-group-id: -----> {0}:
hold-active: -----> {false}:
weight: -----> {nopreference}:
card-line-type: -----> {unknowntype}: adsl-pots-pv-rng-itm
card-atm-configuration: -> {notapplicable}:
card-line-voltage: -----> {not-used}:
maxvpi-maxvci: -----> {notapplicable}:
card-init-string: -----> {}:
wetting-current: -----> {disabled}:
pwe-timing-mode: -----> {none}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

- 3 Verify the card by entering **slots**:

```
zSH> slots
MXK 819
Uplinks
  a:*MXK EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK EIGHT GIGE (NOT_PROV)
Cards
  1: MXK ADSL-48-A Bonded (RUNNING)
  2: MXK ADSL-48-A Bonded (RUNNING)
  3: MXK ADSL-48-A Bonded (RUNNING)
  4: MXK ADSL-48-A Bonded (RUNNING)
  5: MXK ADSL-48-A Bonded (RUNNING)
```

```

6: MXK ADSL-48-A Bonded (RUNNING)
7: MXK ADSL-48-A Bonded (RUNNING)
8: MXK ADSL-48-A Bonded (RUNNING)
9: MXK ADSL-48-A Bonded (RUNNING)
10: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (RUNNING)
11: MXK ADSL-48-A Bonded/with 900 Ohm Splitter (RUNNING)
13: MXK ADSL-48-A Bonded/with 900 Ohm Splitter (RUNNING)
15: MXK ADSL-48-A Bonded/with 600 Ohm Splitter (RUNNING)
17: MXK ADSL-48-A Bonded/with 600 Ohm Splitter (RUNNING)

```

4 Verify the **card-profile** for the ADSL card:

```

zSH> get card-profile 1/10/10202
card-profile 1/10/10202
sw-file-name: -----> {mxlc48aadslbond.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {adsl-pots-pv-rng-itm}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}

```

5 View card information including the state of the card and how long the card has been running:

```

zSH> slots 10
MXK 819
Type           : MXK ADSL-48-A Bonded
Sub-Type       : with Packet Voice POTS, RNG, ITM
Card Version   : 800-02968-01-B
EEPROM Version : 1
Serial #       : 4069337
CLEI Code      : No CLEI
Card-Profile ID : 1/10/10202
Shelf          : 1
Slot           : 10
ROM Version    : MXK 2.0.100
Software Version: MXK 2.1.208
State          : RUNNING
Mode           : FUNCTIONAL
Heartbeat check : enabled
Heartbeat last  : WED AUG 18 16:22:21 2010
Heartbeat resp  : 1229506

```

```

Heartbeat late : 0
Hbeat seq error : 0
Hbeat longest : 10
Fault reset : enabled
Uptime : 13 days, 8 hours, 25 minutes
Packet Voice : Packet mode

```

Internal line testing

The MXK-POTS-72 card and the MXK-ADSL2+-POTS-BCM-48A-RNG-2S card have integrated loop testing functionality, which is same as in the MTAC Enhance card or the TAC-ITM card.

For the detail testing procedures, refer to [MXK Test Access Cards, page 1523](#).

ADSL2+ bond 48-port card configuration

[Table 137](#) shows the card type and software image for the ADSL2+ bond cards on the MXK.

Table 137: MXK ADSL2+ bond card types

Card	Type	Name of software image
MXK-ADSL2+-BCM-48A	10202	<i>mxlc48aadslbond.bin</i>
MXK-ADSL2+-SPLTR600-BCM-48-2S	10202	<i>mxlc48aadslbond.bin</i>
MXK-ADSL2+-SPLTR900-BCM-48-2S	10202	<i>mxlc48aadslbond.bin</i>

The types of 48-port ADSL2+ bonded cards on the MXK are:

- **unknowntype** (default)
- **adsl-pots-pv** (ADSL with VoIP)
- **adsl-splitter** (ADSL with splitters)

Adding ADSL2+ cards

To add an ADSL2+ card to the system:

- 1 Install the ADSL2+ card in the desired line card slot.
- 2 Create a **card-profile** for the card:

```
zSH> card add 1
```

After performing a **card add** in a slot, the slot resets and begins downloading the software image from the flash card. This could take a few moments.

When the card has finished loading, a log message similar to the following is displayed (if logging is enabled):

```
zSH> Card in slot slot-number changed state to RUNNING
```

3 Verify the card by entering slots:

```

zSH> slots
MXK 819
Uplinks
  a:*MXK EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK EIGHT GIGE (NOT_PROV)
Cards
  1: MXK ADSL-48-A Bonded (RUNNING)
  2: MXK ADSL-48-A Bonded (RUNNING)
  3: MXK ADSL-48-A Bonded (RUNNING)
  4: MXK ADSL-48-A Bonded (RUNNING)
  5: MXK ADSL-48-A Bonded (RUNNING)
  6: MXK ADSL-48-A Bonded (RUNNING)
  7: MXK ADSL-48-A Bonded (RUNNING)
  8: MXK ADSL-48-A Bonded (RUNNING)
  9: MXK ADSL-48-A Bonded (RUNNING)
 10: MXK ADSL-48-A Bonded (RUNNING)
 11: MXK ADSL-48-A Bonded/with 900 Ohm Splitter (RUNNING)
 13: MXK ADSL-48-A Bonded/with 900 Ohm Splitter (RUNNING)
 15: MXK ADSL-48-A Bonded/with 600 Ohm Splitter (RUNNING)
 17: MXK ADSL-48-A Bonded/with 600 Ohm Splitter (RUNNING)

```

4 Verify the card-profile for the ADSL card:

```

zSH> get card-profile 1/1/10202
card-profile 1/1/10202
sw-file-name: -----> {mx1c48aads1bond.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {unknowntype}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}

```

5 View card information including the state of the card and how long the card has been running:

```

zSH> slots 1
MXK 819
Type           : MXK ADSL-48-A Bonded
Card Version   : 800-02775-01-B
EEPROM Version : 1

```

```

Serial #           : 2368431
CLEI Code          : No CLEI
Card-Profile ID    : 1/1/10202
Shelf              : 1
Slot               : 1
ROM Version        :
Software Version   : release 1.16
State              : RUNNING
Mode               : FUNCTIONAL
Heartbeat check    : enabled
Longest hbeat      : 18616
Fault reset        : enabled
Uptime             : 18 hours, 45 minutes

```

View additional card information

View the EPROM version of the card:

```

zSH> eeshow card 1
EEPROM contents: for slot 1
EEPROM_ID      : 00 -- CARD
Version        : 01
Size           : 054
CardType       : 10202 -- MXLC48AADSLBOND
CardVersion    : 800-02775-01-B
SerialNum      : 02368431
ShelfNumber    : 00001
CLEI Code      : No CLEI
Cksum          : 0x1F8E

```

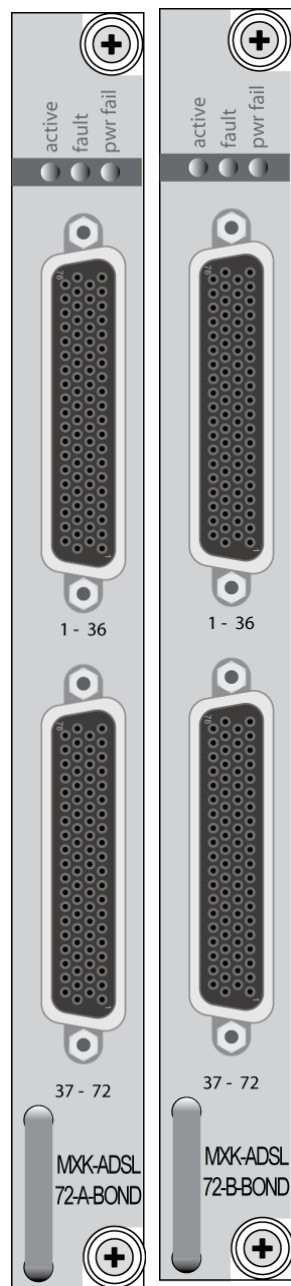
View the EPROM version of the daughter card:

```

zSH> eeshow 1d 9
EEPROM contents: for slot 9
EEPROM_ID      : 01 -- 1DAUGHTER
Version        : 01
Size           : 022
CardType       : 05080 -- MALC_ADSL_48_ANNEXA_M_BOND
CardVersion    : 800-02451-01-A
SerialNum      : 01578673
ShelfNumber    : 00001
CLEI Code      : No CLEI
Cksum          : 0xB00D

```

ADSL2+ bond 72-port card overview



MXK-ADSL2+-BCM-72A

MXK-ADSL2+-BCM-72B

These cards are a single slot card that supports ADSL2+ Annex A/M or ADSL2+ Annex B. The standards supported are ANSI T1.413 Issue 2, G.992.1 (G.dmt), G.992.2 (G.lite), and ADSL2+ (G.992.5) standards.



Caution: Use caution when inserting and removing the ADSL2+ cards to and from the MXK chassis. Be careful not to damage the caps on the board.

ADSL2+ bond 72-port card specifications

Table 135 provides the specifications for the MXK-ADSL2+-BCM-72A

Table 138: ADSL-72-BCM cards specifications

Specification	Description
Size	Single slot
Density	72 ports ADSL
Connectors	Two HDD78 78-pin connectors
Standards supported	ANSI T1.413.2 G.992.1 (G.DMT) G.992.2 (G.Lite) G.992.3 and G.992.4 (ADSL2) G.992.5 (ADSL2+)
Line characteristics	Annex A supported (ADSL2+ over POTS) Annex B supported (ADSL2+ over ISDN) Annex A/M supported (ADSL2/ADSL2+) Fast Path or Interleaved mode supported on a per port basis Fast Retrain supported

Table 138: ADSL-72-BCM cards specifications (Continued)

Specification	Description
Supported train rates	T1.413 (fast and interleaved): - up to 8192 Kbps downstream - up to 1024 Kbps upstream G.lite (interleaved only) - up to 1536Kbps downstream - up to 512Kbps upstream G.DMT (interleave): - up to 12288Kbps downstream - up to 1331.2Kbps upstream G.DMT (fast): - up to 8192Kbps downstream - up to 1024Kbps upstream ADSL2: - Annex A: up to 12288 Kbps downstream - Annex A: up to 1331.2 Kbps upstream - Annex B: 32 Kbps to 28 Mbps downstream 32 Kbps to 1.2 Mbps upstream ADSL2+: - Annex A: up to 28672Kbps downstream - Annex A: up to 1024Kbps upstream - Annex B: 32 Kbps to 28 Mbps downstream - Annex B: 32 Kbps to 1.2 Mbps upstream - Annex M: up to 24576Kbps downstream - Annex M: up to 3584Kbps upstream In the case of Annex M, the upstream will vary depending on how many bins are allocated from the downstream to the upstream. Typically the best case for increasing the upstream bandwidth is approximately 2252.8Kbps.
Power ADSL2+	80 Watts nominal 120 W maximum total. This is at maximum distance with all ports trained at ADSL2+ rates.
Chip set	Broadcom

ADSL2+ bond 72-port card configuration

Each card installed in the system must have a **card-profile**. The line card type determines the parameter settings in the **card-profile** and the software image

for the card. Performing a **card add** <slot #> automatically creates the **card-profile** for the card with the correct software image and settings.

Table 137 shows the card type and software image for the ADSL2+ bond cards on the MXK.

Table 139: MXK ADSL2+ bond card types

Card	Type	Name of software image
MXK-ADSL2+-BCM-72A	10212	<i>mxlc72aadslbond.bin</i>

The types of 72-port ADSL2+ bonded cards on the MXK are:

- **unknowntype** (default)
- **adsl-pots-pv** (ADSL with VoIP)
- **adsl-splitter** (ADSL with splitters)

Adding ADSL2+ cards

To add an ADSL2+ card to the system:

- 1 Install the ADSL2+ card in the desired line card slot.
- 2 Create a **card-profile** for the card:

```
zSH> card add 2
```

After performing a **card add** in a slot, the slot resets and begins downloading the software image from the flash card. This could take a few moments.

When the card has finished loading, a log message similar to the following is displayed (if logging is enabled):

```
zSH> Card in slot slot-number changed state to RUNNING
```

- 3 Verify the card by entering **slots**:

```
zSH> slots
MXK 819
```

Uplinks

a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)

b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)

Cards

```
2: MXK ADSL-72-A Bonded (RUNNING)
4: MXK ADSL-48-A Bonded (NOT_PROV)
18: MXK 8 PORT GPON (RUNNING)
```

- 4 Verify the **card-profile** for the ADSL card:

```
zSH> get card-profile 1/2/10212
```

```

card-profile 1/2/10212
sw-file-name: -----> {mxlc72aads1bond.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {unknowntype}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}

```

5 View card information including the state of the card and how long the card has been running:

```

zSH> slots 2
MXK 819
Type           : MXK ADSL-72-A Bonded
Card Version   : 800-02804-03-A
EEPROM Version : 1
Serial #       : 4966242
CLEI Code      : No CLEI
Card-Profile ID : 1/2/10212
Shelf          : 1
Slot           : 2
ROM Version    : MXK 2.1.211
Software Version: release_2.1
State          : RUNNING
Mode           : FUNCTIONAL
Heartbeat check : enabled
Heartbeat last  : MON OCT 11 20:27:14 2010
Heartbeat resp  : 506349
Heartbeat late  : 0
Hbeat seq error : 0
Hbeat longest   : 6
Fault reset     : enabled
Uptime         : 5 days, 20 hours, 39 minutes

```

View additional card information

View the EPROM version of the card:

```

zSH> eeshow card 2
EEPROM contents: for slot 2
EEPROM_ID      : 00 -- CARD
Version        : 01

```

Size : 054
CardType : 10212 -- MXLC72AADSLBOND
CardVersion : 800-02804-03-A
SerialNum : 04966242
ShelfNumber : 00001
CLEI Code : No CLEI
Cksum : 0x0EOF

ADSL2+ on the MXK

This section describes how ADSL functions on the MXK including:

- [ADSL2+ overview, page 1202](#)
- [ADSL2+ transmission modes, page 1203](#)
- [ADSL2+ rate adaptation, page 1203](#)
- [Advanced ADSL2+ configurations on the MXK, page 1204](#)

ADSL2+ overview

MXK ADSL2+ interfaces provide a standards-based, high-speed DSL interface between the MXK and CPE devices.

Asymmetric Digital Subscriber Line (ADSL) is a type of DSL that uses existing telephone lines to solve the issue of first mile connection. It is more expensive to dig trenches and lay fiber than it is to deploy ADSL technology over twisted wire pairs of existing telephone lines.

This is possible because voice signals use the portion of the frequencies which can be sent over a twisted wire pair below 4kHz and ADSL uses the portion of the frequencies above 25kHz.

ADSL is asymmetric because the data flow is greater in one direction. The range of frequencies used by ADSL is separated into two frequency bands — the upstream band to the central office and the downstream band to the end user. The downstream band is larger, hence downloads to a home computer are faster than uploads.

Signals sent down copper wire may be impaired by distance from the central office, noise on the wire, and radio interference from AM radio stations.

ADSL devices can adjust to signal conditions to achieve the highest possible speeds, so usually no adjustment is needed. The ability to adjust to signal conditions is called “training.” The default settings used by ADSL cards on the MXK are suitable in most cases, although configuration options are available for fine tuning when necessary.

Configurable ADSL2+ options on the MXK pertain to the accuracy of transmission packets and overall throughput as shown in [Table 140](#).

Table 140: Configurable ADSL2+ options

Option	Description
Signal to Noise Ratio	Provides a mechanism to adjust the robustness of the ADSL Link and hence the speed.
Transport Mode	Defines how packets are sent down the line. Fast provides a simple contiguous message which does not require much processing time to disassemble and reassemble packets. Interleaved provides greater protection from short bursts of noise that can result in lost packets

Table 140: Configurable ADSL2+ options (Continued)

Option	Description
Bonding	Bonding is the ability to have multiple ports work together, so they appear as one larger pipe. ADSL bonding allows combining two ports.

ADSL2+ transmission modes

Table 141 describes the transmission modes MXK ADSL2+ cards support.

Table 141: Supported transmission modes

Transmission Mode	Description
ADSL2	The modem negotiates rates up to 12 Mbps downstream and 3.5 Mbps upstream. G.992.3 ADSL2.
ADSL2plus	The modem negotiates rates up to 28 Mbps downstream and 1 Mbps upstream (Annex M allows upstream up to 3.5 Mbps) G.992.5 ADSL2+.
Autonegotiate	The modem automatically negotiates all supported transmission modes.
Full rate	Full rate T1 ADSL modem. This is used for connecting to full rate T1.413 issue 2 modems.
G.dmt	G.dmt is a higher-bandwidth variant of G.lite that provides for downstream speeds of up to 8160 Kbps. G.dmt is defined in ITU specification G.992.1.
G.lite	The modem negotiates rates up to 1536 Kbps upstream and 512 Kbps downstream (G.992.2).

ADSL2+ rate adaptation

The ADSL2+ cards support rate adaptation, which allow them to respond to changing line conditions by adjusting the line rate. At startup, ADSL2+ ADSL2+ modems may negotiate a data rate. The **rateMode** parameter allows the selection of three types of rate adaption:

- **fixed:** rate is fixed at the max configured rate.
- **adaptatstartup:** rate is set to the best possible speed (between min and max) during training and does not change afterward.
- **adaptatruntime:** rate is set to the best possible speed (between min and max) during training and can change afterward based on changing conditions

The default option is **adaptatruntime**, so the rate can change based on changing conditions.

Advanced ADSL2+ configurations on the MXK

This section describes configuring signal-to-noise (SNR) parameters and describes:

- [Fine tuning ADSL2+ video performance, page 1204](#)
- [Seamless Rate Adaptation, page 1207](#)
- [Transport mode: fast or interleaved, page 1209](#)

ADSL2+ modems use signal-to-noise ratio (SNR) measurements to adjust signal transmission to achieve greater performance. The Zhone default settings for SNR parameters normally provide an excellent throughput rate for most applications.

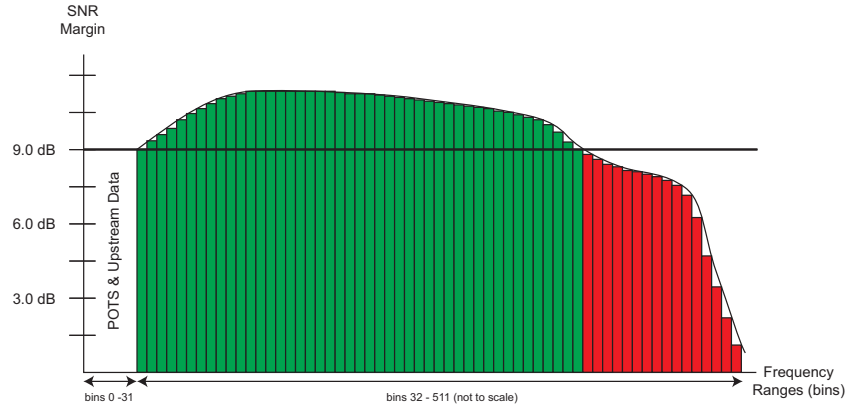
Fine tuning ADSL2+ video performance

The parameters for tuning performance may be adjusted for video. These parameters are part of a complex system. Before making changes to the default settings you should understand the SNR parameters and how they work together.

This section describes guidelines for adjusting SNR settings and will not be correct for every deployed line. Subscribers with “noisy” lines may need to have their ADSL2+ parameters adjusted so that the train rates are high enough to meet the service bandwidth requirements. This section discusses how adjusting SNR Margins can increase train rates while keeping errors on the line to a minimum.

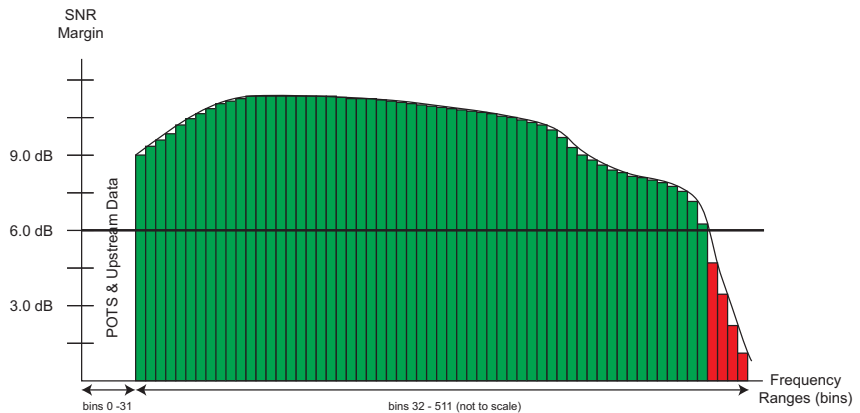
SNR compares the level of the desired signal to the level of background noise. The better the signal and the less obtrusive the background noise, the higher the ratio. The lower the SNR, the greater effect noise will have on the ADSL2+ signal. Noise is anything that will corrupt the sent signal and is typically from AM radio transmissions, although poor physical connections, deformities in the line, transformers, and even appliances can also introduce noise.

If it weren't for noise you could set the SNR very high and not be worried about signal degradation. Unfortunately in the field, not all ADSL2+ lines will train when the SNR margin target is set high. With the target SNR margin set too high, the ADSL2+ training algorithm would try to make the line so clean (no noise) that the train rate would be very low and not capable of supporting the services sold.

Figure 153: Bins shown with SNR Margin set to 9.0 dB

The frequency bands on DSL lines are segmented into small frequency ranges called bins or tones. These small ranges make it so the frequency can be sampled to judge the value. There are 512 bins in a signal. The voice and upstream data traffic use only a small portion (bins 0-31) and are not relevant to this discussion. Bins 32-511 are used for downstream data traffic.

If the SNR is dropped to a lower rate with the same signal to noise ratio, more of the sampled bins are used.

Figure 154: Bins shown with SNR Margin set to 6.0 dB

[Figure 153](#) and [Figure 154](#) show a snapshot of the signal.

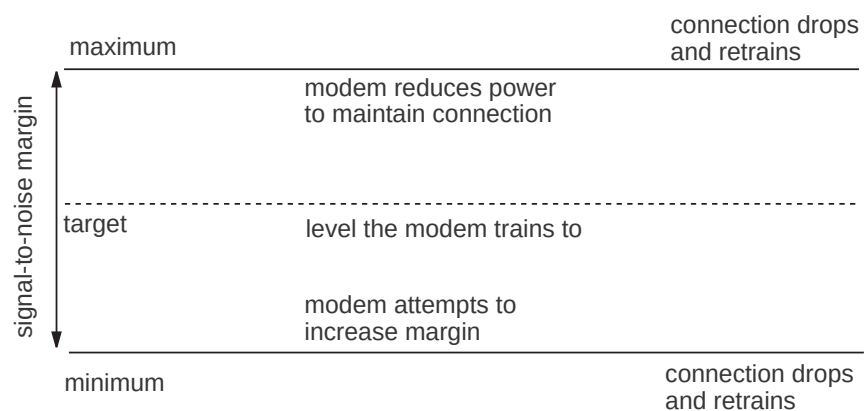
[Table 142](#) describes the three parameters in the **adsl-co-profile** and the **adsl-cpe-profile** which define the training speeds.

Table 142: adsl-co-profile and adsl-cpe-profile parameters

Parameter	Profile	Description
targetSnrMgn	adsl-co-profile adsl-cpe-profile	The Target SNR Margin (targetSnrMgn) is the SNR Margin targeted when training. Values are from 0 to 310 in tenths of dBs. A value of 60 would mean 6.0 dB SNR Default: 0 Recommendation for Video: 60
maxSnrMgn	adsl-co-profile adsl-cpe-profile	Maximum SNR Margin (maxSnrMgn) is the maximum SNR Margin allowed on the link before a retrain is forced. Values are from 0 to 310 in tenths of dBs. A value of 150 would mean 15.0 dB SNR. Default: 0 Recommendation for Video: 150
minSnrMgn	adsl-co-profile adsl-cpe-profile	Minimum SNR Margin (minSnrMgn) is the minimum SNR Margin allowed on the link before a retrain is forced. Values are from 0 to 310 in tenths of dBs. Default: 0 Recommendation for Video: 30

SNR performance is monitored to maintain a bit error rate (BER) of 10^{-7} or better. The minimum margin is the floor at which the modem will maintain a connection. The maximum margin is the ceiling for power cutback. The target margin is the lowest margin that the modem tries to achieve when training and adapting. [Figure 155](#) shows single-to-noise margin values.

Figure 155: Signal-to-noise margins



These three values alone allow the ADSL2+ line to train to a maximum rate given the target SNR Margin value. That initial train rate would remain unless the SNR Margin moves beyond the Minimum or Maximum SNR Margin. At that time the link is forced to retrain.

The system will try to attain the target signal-to-noise margin when training. If the line reaches the maximum bit rate and the actual margin is below the maximum margin, the line operates normally. If the margin rises above the target margin, the modem drops the connection and retrains once, then drops the power to enforce the maximum margin.

If, after a connection is made, the margin drops below the target margin, the modem attempts to increase the margin. If the minimum margin cannot be kept, the modem drops the connection and retrains.

Note within the above table are the Zhone recommended values for video. These SNR Margin values may not be appropriate on every link, but based on Zhone's testing they result in high train rates and low error rates on most lines. For loops with excessive noise which prevents the necessary data rate for video services, adjust the targetSnrMgn to 60. Lowering the Target SNR Margin should allow the line to train higher.

Retraining the signal takes a considerable amount of time (as much as 30 seconds). An ADSL2+ feature Seamless Rate Adaption (SRA) can make more minute adjustments within the minimum and maximum SNR margins without the end user being aware of the rate changes or time to retrain.

Seamless Rate Adaptation

After an ADSL2+ link trains the noise conditions on the line could improve. Seamless Rate Adaptation allows the ADSL2+ link to take advantage of the lower noise and will increase the rate of the link without the need for a retrain. SRA may also reduce the rate on the line when noise levels increase slightly.

The Upshift SNR Margin (**upshiftSnrMgn**) and Downshift SNR Margin (**downshiftSnrMgn**) are used to determine the values to conduct the rate adaptation by adding or removing bins to stay at the target SNR. Time parameters work with the Upshift and Downshift SNR Margins, Minimum Upshift SNR Time (**minUpshiftTime**) and Minimum Downshift Time (**minDownshiftTime**) for the Central Office (**adsl-co-profile**) and Minimum Upshift SNR Margin (**minUpshiftSnrMgn**) and Minimum Upshift SNR Margin (**minDownshiftSnrMgn**), which are also for the modem side of the connection (**adsl-cpe-profile**). The CO profile and CPE profile use different names for the similar parameter because the CO is the head of the connection and accepts requests from the CPE devices, but can determine the minimum time to conduct the SRA.

All of these parameters work together in a system. When the SNR rises above the Upshift SNR Margin and stays there for a specified amount of time (from the minUpshiftTime and minUpshirtSnrMgn) it is assumed that the noise level has improved and the rate is allowed to increase.

As the SNR moves below the Downshift SNR Margin value and stays there for a specified amount of time, the noise level has increased and the current noise level can not sustain the current downstream rate without increased errors so the rate is decreased. The increases and decreases in rate are done "seamlessly" and without the need to retrain the line.

Figure 156: SNR Margins working as a system

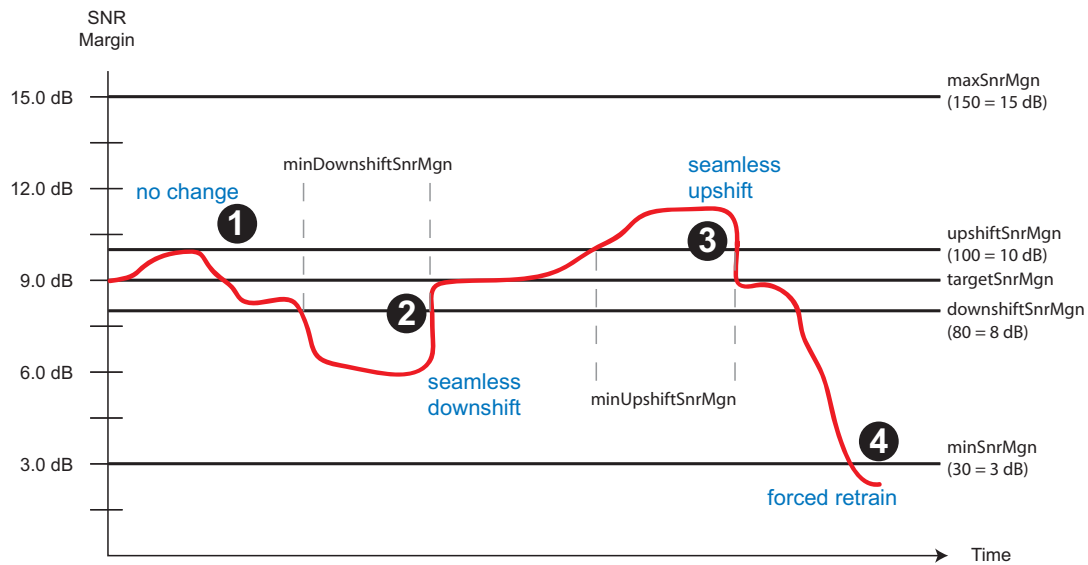


Figure 156 shows how the five SNR Margin parameters work as a system to ensure the best train rate possible within the given parameters. The red line represents how the SNR changes over time. The SNR Margin increases, but does not move past the Upshift SNR Margin at (1) so the train rate remains the same. At (2) on the graph the SNR Margin has dipped below the Downshift SNR Margin and stays below downshiftSnrMgn longer than the minimum downshift margin time. This situation results in a removal of bins in order to return to the Target SNR Margin. This change is a seamless decrease in the data rate from the user's perspective. The SNR Margin then rises and moves above the Upshift SNR Margin for longer than **minUpshiftSnrMgn** period resulting in a seamless increase in the rate at (3). In this situation bins are added to get back to the Target SNR Margin. The SNR then moves down quickly below the Min SNR Margin which forces a retrain at (4).

Note that each parameter plays an important role in the training of the ADSL2+ line. The SNR margins should always have **maxSnrMgn > upshiftSnrMgn > targetSnrMgn > downshiftSnrMgn > minSnrMgn**. If the Minimum and Maximum SNR Margins are brought too close to the target SNR Margin on a line which has changing SNR, there could be excessive retraining. If the SRA values Upshift SNR Margin and Downshift SNR Margin are too close to the Maximum and Minimum SNR values, SRA will not be useful, the line will retrain by the Minimum and Maximum SNR values.

Setting the SRA shift values too high for the upshift and too low for the downshift makes the probability of an SRA shift unlikely. A good configuration rule for determining **downshiftSnrMgn** and **upshiftSnrMgn**:

$$\text{downshiftSnrMgn} = \text{targetSnrMgn} + 10$$

$\text{upshiftSnrMgn} = \text{targetSnrMgn} - 10$

SRA is only supported in the downstream data direction and the CPE is the controlling device for the feature. SRA is configured in the **adsl-cpe-profile**. Changes to the **adsl-co-profile** are ignored.

There are two timers used to space SRA events. The downstream (CO to CPE) SRA timers are located in the **adsl-cpe-profile**. The SRA timers are in units of seconds so a value of 60 means an SRA event can only occur every 60 seconds.

Zhone's recommended settings are:

$\text{minUpshiftSnrMgn} = 30$

$\text{minDownshiftSnrMgn} = 30$

The SRA timers start after the first SRA action which means that an SRA rate shift can occur immediately after initial train up.

For SRA to operate the CPE must support SRA and must have SRA enabled.

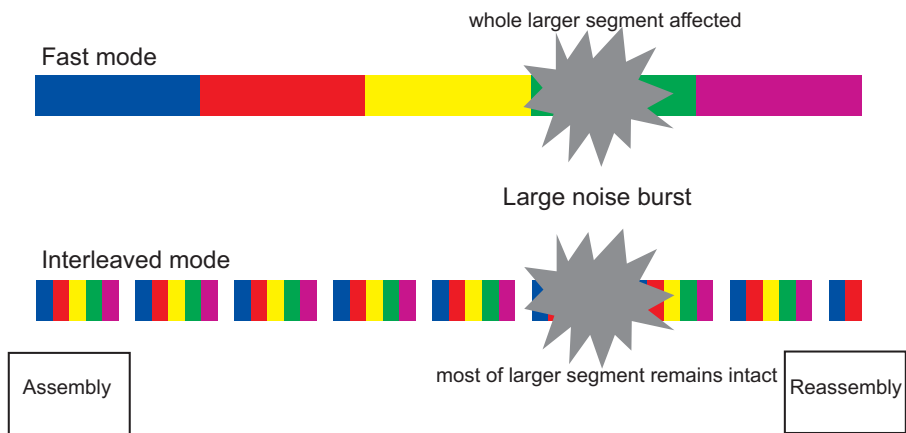
Transport mode: fast or interleaved

ADSL2+ operates in one of two modes: fast or interleaved. In fast mode, data packets are placed on the ADSL2+ line contiguously providing data in a sequential stream. The other end of the ADSL2+ line removes the data packets in order, then moves them up the protocol stack for processing. In interleaved mode, the data packets are broken into smaller segments, then sent down the line.

The advantage of fast mode is that the data is streamed directly without disassembly and assembly processing. However, a short burst of noise on the line can corrupt more errors than the far end device can correct. ADSL2+ modems have the ability to correct errors; however error correction works well when there are a small number of errors to correct. Too many bit errors in a packet can mean the errors can not be corrected and result in lost data packets. Lost data packets require that the same data packet be retransmitted.

With smaller segments use interleaving mode, the segments may be intermixed with segments from other packets before being placed on the ADSL2+ line. If a burst of noise causes corruption, each entire larger pre-disassembled packet is not affected as much, but the smaller pieces which could belong to several different packets are affected. Because only smaller segments of the larger packet are affected, error correction is more likely to have enough information to build the packet correctly on reassembly. The data packets handed up the stack will have no issues.

Figure 157: Fast and interleaved mode



The drawback with Interleaving is that the process of interleaving the small data blocks and reassembling the data packets at the far end introduce some delay and lowers the data rate.

It is recommended to use Fast mode with data applications.

Interleaved mode should be used with video applications. Video applications usually do not support retransmissions. If a data packet is corrupted it is discarded and will not be retransmitted so it is important that as many packets as possible arrive in good condition.

Fast configuration notes

On the MXK, fast and interleave modes are configured in the `adslChannelMode` field in the `adsl-profile`.

[Table 143](#) shows the settings are used for fast operations.

Table 143: Fast operation settings

Parameter	Profile	Description
fastMinTxRate	adsl-co-profile adsl-cpe-profile	Minimum transmit rate in bits per second (bps) for channels configured for fast transmission mode. fastMinTxRate must be less than fastMaxTxRate. Default: 0
fastMaxTxRate	adsl-co-profile adsl-cpe-profile	Configured maximum transmit rate (bps) for ADSL Fast channels. fastMaxTxRate must be greater than fastMinTxRate. Default: 8460Kbps

Table 143: Fast operation settings (Continued)

Parameter	Profile	Description
threshFastRateUp	adsl-co-profile adsl-cpe-profile	Not currently used. The change in the configured rate that causes the system to send an adslAtucRateChangeTrap . The system sends a trap whenever: ChanCurrTxRate <= ChanPrevTxRate minus the value of this object. A value of 0 disables the trap. Default: 0
threshFastRateDown	adsl-co-profile adsl-cpe-profile	Not currently used. The change in the configured rate that causes the system to send an adslAturRateChangeTrap . The system sends a trap whenever: ChanCurrTxRate <= ChanPrevTxRate minus the value of this parameter. Default: 0

Interleaved configuration notes

[Table 144](#) shows the settings used during interleaved operations.

Table 144: Interleaved operation settings

Parameter	Profile	Description
maxInterleaveDelay	adsl-co-profile adsl-cpe-profile	Defines the mapping (relative spacing) between subsequent interleave input bytes and their placement in the bit stream at the interleave output. Larger numbers provide greater separation between consecutive input bytes in the output bit stream allowing for improved impulse noise immunity, but at the expense of payload latency. Default: 0
interleaveMinTxRate	adsl-co-profile adsl-cpe-profile	Minimum transmit rate (bps) for channels configured for interleaved transmission mode. interleaveMinTxRate must be less than interleaveMaxTxRate . Default: 0
interleaveMaxTxRate	adsl-co-profile adsl-cpe-profile	Maximum transmit rate (bps) for channels configured for interleaved transmission mode. Default: 8160Kbps

Table 144: Interleaved operation settings (Continued)

Parameter	Profile	Description
threshInterleaveRateUp	adsl-co-profile adsl-cpe-profile	Not currently used. The change in the configured rate that causes the system to send an adslAturRateChangeTrap. The system sends a trap whenever: ChanCurrTxRate >= ChanPrevTxRate plus the value of this object. Default: 0
threshInterleaveRateDown	adsl-co-profile adsl-cpe-profile	Not currently used. The change in the configured rate that causes the system to send an adslAtucRateChangeTrap. The system sends a trap whenever: ChanCurrTxRate <= ChanPrevTxRate minus the value of this object. Default: 0

ADSL2+ interface configuration

This section explains ADSL2+ configuration on the MXK. It contains the following sections:

- [ADSL2+ interface overview, page 1213](#)
- [View adsl-profile parameter defaults, page 1214](#)
- [View adsl-co-profile parameter defaults, page 1217](#)
- [View adsl-cpe-profile parameter defaults, page 1226](#)
- [Upstream and downstream tone ranges, page 1234](#)
- [Configure ADSL2+ profiles for Annex M in fast mode, page 1235](#)
- [Configure ADSL2+ profiles for Annex M in interleaved mode, page 1238](#)
- [Configure ADSL2+ profiles for G.lite, page 1241](#)
- [Configure ADSL2+ profiles to cap train rates, page 1244](#)
- [Configure ADSL2+ S=1/2, page 1249](#)
- [Configure Broadcom Phy-R™ parameters, page 1255](#)
- [Configure G.INP parameters, page 1257](#)
- [ADSL2+ statistics, page 1283](#)

ADSL2+ interface overview

ADSL2+ configuration consists of three profiles:

- **adsl-profile**
- **adsl-co-profile**
- **adsl-cpe-profile**

[Table 145](#) summarizes the **update** commands used to configure ADSL2+ interfaces on the MXK:

Table 145: Commands to configure ADSL2+ interfaces

Action	Command
Configure the ADSL2+ interface. See Configure ADSL2+ profiles for Annex M in fast mode on page 1235 .	<code>update adsl-profile shelf/slot/port</code> where <i>port</i> is from 1 to 48
Configure the downstream interface. See Configure ADSL2+ profiles for Annex M in fast mode on page 1235 .	<code>update adsl-co-profile shelf/slot/port</code>

Table 145: Commands to configure ADSL2+ interfaces (Continued)

Action	Command
Configure the upstream interface. See Configure ADSL2+ profiles for Annex M in fast mode on page 1235	<code>update adsl-cpe-profile shelf/slot/port</code>
Configure ADSL S=1/2. See Configure ADSL2+ S=1/2 on page 1249 .	<code>update adsl-profile shelf/slot/port</code> <code>update adsl-co-profile shelf/slot/port</code>

View adsl-profile parameter defaults

The **adsl-profile** defaults are appropriate for most applications. When necessary, the **adsl-profile** parameters can be updated.

Viewing the adsl-profile defaults

View the **adsl-profile** parameters and their default values.

1 View the ADSL2+ cards:

```
zSH> slots
MXK 819
Uplinks
  a:*MXK EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK EIGHT GIGE (NOT_PROV)
Cards
  1: MXK ADSL-48-A Bonded (RUNNING)
  2: MXK ADSL-48-A Bonded (RUNNING)
  11: MXK ADSL-48-A Bonded/with 900 Ohm Splitter (RUNNING)
  17: MXK ADSL-48-A Bonded/with 600 Ohm Splitter (RUNNING)
```

2 View the **adsl-profile** parameters and values:

```
zSH> show adsl-profile
adslLineConfProfile:-----> {260}
adslAlarmConfProfile:-----> {260}
adslTrellisModeEnabled:-----> true false
adslNTRModeEnabled:-----> true false
adslTransmissionMode:-----> autonegotiatemode fullratemode glitemode
t1mode gdmtdmode ghsmode adsl2mode adsl2plusmode reachdslmode
adslChannelMode:-----> fastonly interleavedonly fastandinterleaved
fastorinterleaved
adslMaxDownstreamToneIndex:-----> {6 - 1023}
adslMinDownstreamToneIndex:-----> {6 - 1023}
adslMaxUpstreamToneIndex:-----> {1 - 63}
adslMinUpstreamToneIndex:-----> {1 - 63}
adslPotsBypassRelayMaxDuration:-> {1 - 300}
adslLineDMTConfMode:-----> echocancel freqdivmux
adslAnnexMModeEnabled:-----> true false
adslAnnexMPsdMask:-----> eu64 eu60 eu56 eu52 eu48 eu44 eu40 eu36
eu32 all
adslAnnexJModeEnabled:-----> true false
```

Table 146 defines **adsl-profile** parameters values.

Table 146: adsl-profile parameter definitions

Parameter	Description
adslLineConfProfile	Read only.
adslAlarmConfProfile	Read-only.
adslTrellisModeEnabled	Enables or disables trellis mode.
adslTransmissionMode	ADSL transmission mode. Supported values: Values: adsl2mode The modem negotiates rates up to G.992.3 and G.992.4 ADSL2. ADSL2plusmode The modem negotiates rates up to G.992.5 (ADSL2+). autonegotiatemode : automatically negotiates all supported transmission modes. fullratemode : automatically negotiates full rate modes (G.dmt and T1 mode). G.dmt has priority over T1 mode. glitemode : G.lite. Supports only interleave mode. t1mode : Full rate T1 gdmtdmode : G.dmt ghsmode : the modem negotiates only G.dmt and G.lite modes. G.dmt has priority over G.lite. reachdslmode the modem negotiates only reach DSL mode. Default: autonegotiatemode
adslChannelMode	Specifies the channelization of the ADSL line. Supported values: Values: fastonly No impulse noise protection, but lowest possible latency. Recommended only where lowest possible latency is required (for example, gaming) interleavedonly Better impulse noise protection with higher latency. Recommended for all voice, video, and/or data deployments. Default: fastonly
adslMaxDownstreamToneIndex	Specifies the maximum downstream active tone. Values: 32 (128KHz) to 511 (2044KHz) Each value represents 4KHz. Default: 255 Changing this value causes the DSL modems to retrain.

Table 146: adsl-profile parameter definitions (Continued)

Parameter	Description
adslMinDownstreamToneIndex	Specifies the minimum downstream active tone. Values: 32 (128KHz) to 255 (1020KHz) Each value represents 4KHz. Default: 511 - Changing this value causes the DSL modems to retrain. - For Annex B and Annex M configurations, this value should be set to 64.
adslMaxUpstreamToneIndex	Specifies the maximum upstream active tone. Values: 6 (24KHz) to 30 (120KHz) Each value represents 4KHz. Default: 3 - Changing this value causes the DSL modems to retrain. - For Annex B and Annex M configurations, this value should be set to 63.
adslMinUpstreamToneIndex	Specifies the minimum upstream active tone. Values: 6 (24KHz) to 30 (120KHz) Each value represents 4KHz. Default: 6 Changing this value causes the DSL modems to retrain.
adslPotsBypassRelayMaxDuration	Not currently used. The maximum duration in seconds that an ADSL POTS low-pass filter bypass relay will remain active (closed). The relay will automatically return a line back to normal (open) mode when this timer has expired. Values: 1 to 300 Default: 60 Only valid for ADSL-SPLTR-32 cards.
adslLineDMTConfMode	Selects whether there is overlap of ADSL Discrete Multi-Tone (DMT) frequency bins. Values: echoCancel overlap of DMT frequency bins. Only supported by g.dmt Annex A. freqDivMux no overlap of DMT frequency bins. Separates downstream and upstream transmissions. Default: freqdivmux
annexMModeEnabled	Specifies whether annex M mode is enabled. This parameter can only be set to true when the adslTransmissionMode parameter is set to autonegotiate, adsl2mode, or adsl2plusmode. Default: false

Table 146: adsl-profile parameter definitions (Continued)

Parameter	Description
adslAnnexMPsdMask	eu64 eu60 eu56 eu52 eu48 eu44 eu40 eu36 eu32 all
adslAnnexJModeEnabled	Default: false

View adsl-co-profile parameter defaults

The ADSL2+ downstream interface is **adsl-co-profile** which defines the downstream behavior.

Viewing the adsl-co-profile defaults

View the **adsl-co-profile** parameters and their default values.

1 View the ADSL2+ cards:

```
zSH> slots
MXK 819
Uplinks
  a:*MXK EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK EIGHT GIGE (NOT_PROV)
Cards
  1: MXK ADSL-48-A Bonded (RUNNING)
  2: MXK ADSL-48-A Bonded (RUNNING)
 11: MXK ADSL-48-A Bonded/with 900 Ohm Splitter (RUNNING)
 17: MXK ADSL-48-A Bonded/with 600 Ohm Splitter (RUNNING)
```

2 View the **adsl-co-profile** default parameter values:

```
zSH> get adsl-co-profile 1/10/1
adsl-co-profile 1/10/1
rateMode: -----> {adaptatruntime}
rateChanRatio: -----> {50}
targetSnrMgn: -----> {60}
maxSnrMgn: -----> {310}
minSnrMgn: -----> {0}
downshiftSnrMgn: -----> {30}
upshiftSnrMgn: -----> {90}
minUpshiftTime: -----> {60}
minDownshiftTime: -----> {60}
fastMinTxRate: -----> {32000}
interleaveMinTxRate: -----> {32000}
fastMaxTxRate: -----> {32736000}
maxInterleaveDelay: -----> {63}
interleaveMaxTxRate: -----> {32736000}
thresh15MinLofs: -----> {0}
thresh15MinLoss: -----> {0}
thresh15MinLols: -----> {0}
thresh15MinLprs: -----> {0}
thresh15MinESs: -----> {0}
threshFastRateUp: -----> {0}
threshInterleaveRateUp: ---> {0}
```

```

threshFastRateDown: -----> {0}
threshInterleaveRateDown: -> {0}
initFailureTrapEnable: ----> {disable}
reachExtendedAdsl2: -----> {enable}
minTxThresholdRateAlarm: --> {0}
minINP: -----> {20}
phyRSupport: -----> {disable}
phyRmaxINP: -----> {0}
phyRminRSoverhead: -----> {0}
phyRRtxRatio: -----> {0}
txPowerAttenuation: -----> {20}
cabMode: -----> {0}
ginpAdslCoSupport: -----> {disable}
ginpAdslCoEtrMax: -----> {32736}
ginpAdslCoEtrMin: -----> {64}
ginpAdslCoNdrMax: -----> {32736}
ginpAdslCoShineRatio: ----> {10}
ginpAdslCoLeftrThreshold: -> {0}
ginpAdslCoMaxDelay: -----> {20}
ginpAdslCoMinDelay: -----> {0}
ginpAdslCoMin: -----> {4}
ginpAdslCoMinRSoverhead: --> {0}
ginpAdslCoReinCfg: -----> {0}
ginpAdslCoReinFreq: -----> {freq120hz}
ginpAdslCoRtxMode: -----> {preferred}

```

Table 147 defines the values for the **adsl-co-profile** parameters.

:

Table 147: adsl-co-profile parameter definitions

Parameter	Description
rateMode	The transmit rate adaptation configured on this modem. Supported values: fixed: The rate is negotiated at startup and remains fixed. Modem speed is determined by the fastMaxTxRate or interleaveMaxTxRate parameters. adaptatstartup: The rate is negotiated at startup and remains fixed. Modem speed is determined by the fastMaxTxRate or interleaveMaxTxRate parameters. If the line is able to support a higher rate, the rate above the minimum is assigned to the available channel (either fast or interleave). adaptatruntime: The rate is negotiated dynamically and can vary between the maximum and minimum configured rates. If the line conditions change during runtime, the line speed is adjusted. Recommended for video. Default: adaptatruntime
rateChanRatio	Configured allocation ratio of excess transmit bandwidth between fast and interleaved channels. Default: 50

Table 147: adsl-co-profile parameter definitions (Continued)

Parameter	Description
targetSnrMgn	Target signal to noise margin (in tenths of dBs). This is the noise margin the modem must achieve with a BER of 10^{-7} or better to successfully complete initialization. Suggested values are 6 dB for data-only or data-voice service and 10 dB for video service with better protection against noise which causes tiling. Default: 60
maxSnrMgn	Maximum acceptable signal/noise margin (in tenths of dBs). If the noise margin rises above this the modem attempts to reduce its power output to optimize its operation. Reduces crosstalk into other ADSL circuits by not transmitting at an unnecessarily high level. For video, suggested values are 31 for both upstream and downstream. Default: 310
minSnrMgn	Minimum acceptable signal to noise margin (in tenths of dBs). If the noise margin falls below this level, the modem attempts to increase its power output. If that is not possible the modem will attempt to re-initialize or shut down. For video, use 2 downstream and 0 upstream and adjust downstream rate proactively just before video degrades. Default: 0
downshiftSnrMgn	Configured Signal/Noise Margin for rate downshift. If the noise margin falls below this level, the modem should attempt to decrease its transmit rate. Default: 0
upshiftSnrMgn	Configured Signal/Noise Margin for rate upshift. If the noise margin rises above this level, the modem should attempt to increase its transmit rate. Default: 0
minUpshiftTime	Minimum time that the current margin is above UpshiftSnrMgn before an upshift occurs. Default: 0
minDownshiftTime	Minimum time that the current margin is below DownshiftSnrMgn before a downshift occurs. Default: 0
fastMinTxRate	Minimum transmit rate (in bps) for channels configured for fast transmission mode. For a CO interface, the range is 32Kbps to 8160Kbps (1536Kbps for G.Lite). Default: 32 Kbps

Table 147: adsl-co-profile parameter definitions (Continued)

Parameter	Description
interleaveMinTxRate	Minimum transmit rate (in bps) for channels configured for interleaved transmission mode. For a CO interface, the range is 32Kbps to 8160Kbps (1536Kbps for G.Lite). Default: 32 Kbps
fastMaxTxRate	Maximum transmit rate (in bps) for channels configured for fast transmission mode. For a CO interface, the range is 32Kbps to 8160Kbps (1536Kbps for G.Lite). Default: 32 Kbps
maxInterleaveDelay	Maximum interleave delay for this channel. Interleave delay applies only to the interleave channel and defines the mapping (relative spacing) between subsequent input bytes at the interleave input and their placement in the bit stream at the interleave output. Larger numbers provide greater separation between consecutive input bytes in the output bit stream allowing for improved impulse noise immunity, but at the expense of payload latency. For video, to maximize protection of downstream signal (where impulse problems occur), minimize round-trip latency by minimizing upstream delay use 1 ms upstream and 16 ms downstream. Values: 0 0.5 ms 1 1 ms 2 2 ms 4 4 ms 8 8 ms 16 16 ms 32 32 ms 63 63 ms Default: 63 ms
interleaveMaxTxRate	Maximum transmit rate (in bps) for channels configured for interleaved transmission mode. For a CO interface, the range is 32Kbps to 8160Kbps (1536Kbps for G.Lite). Default: 32 Mbps
thresh15MinLofs	The number of Loss of Frame Seconds encountered by an ADSL interface within any given 15 minutes performance data collection period, which causes the SNMP agent to send an adslAtucPerfLofsThreshTrap. Default: 0

Table 147: adsl-co-profile parameter definitions (Continued)

Parameter	Description
thresh15MinLoss	The number of Loss of Signal Seconds encountered by an ADSL interface within any given 15 minutes performance data collection period, which causes the SNMP agent to send an adslAtucPerfLossThreshTrap. Default: 0
thresh15MinLols	The number of Loss of Link Seconds encountered by an ADSL interface within any given 15 minutes performance data collection period, which causes the SNMP agent to send an adslAtucPerfLolsThreshTrap. Default: 0
thresh15MinLprs	Not currently used. The number of Loss of Power Seconds encountered by an ADSL interface within any given 15 minutes performance data collection period, which causes the SNMP agent to send an adslAtucPerfLprsThreshTrap. Default: 0
thresh15MinESs	The number of Errored Seconds encountered by an ADSL interface within any given 15 minutes performance data collection period, which causes the SNMP agent to send an adslAtucPerfESsThreshTrap. Default: 0
threshFastRateUp	Not currently used. Applies to 'Fast' channels only. Configured change in rate causing an adslAtucRateChangeTrap. Default: 0
threshInterleaveRateUp	Not currently used. For 'Interleave' channels only. Configured change in rate causing an adslAtucRateChangeTrap. Default: 0
threshFastRateDown	Not currently used. For 'Fast' channels only. Configured change in rate causing an adslAtucRateChangeTrap. Default: 0
threshInterleaveRateDown	Not currently used. For 'Interleave' channels only. Configured change in rate causing an adslAtucRateChangeTrap. Default: 0
initFailureTrapEnable	Not currently used. Enables and disables the InitFailureTrap. This trap controls whether line up or line down traps are sent while the system is booting up. Default: 0

Table 147: adsl-co-profile parameter definitions (Continued)

Parameter	Description
reachextendedAdsl2	Defines whether downstream reach extended ADSL2 (READSL2) operations should be enforced by the ATU-C. Only enabled for ADSL2 and ADSL2+. Values: enable disable Default: enable
minTxThresholdRateAlarm	Enables the CO (downstream) transmission rate threshold value. If the rate falls below this value, the device sends a trap and an alarm. Default: 0
minINP (already used in the case of normal interleaving)	This parameter (already used in the case of normal interleaving) defines the minimal guaranteed impulse noise protection, provided that the available data bandwidth allowed for retransmissions is not exceeded. Default: 20
phyRSupport	Enable to turn on Phy-R™ parameters. Disable to turn off Phy-R™ parameters. Default: 0
phyRmaxINP	This parameter defines the maximum number of consecutive retransmissions that may take place and therefore bounds the maximal jitter due to retransmissions. A default value of zero doesn't bound the number of consecutive retransmissions (that will however never exceed maxDelay * 4 symbols).
minRSoverhead	This new parameter allows to force a minimum amount of RS overhead. This can be used to guarantee a given amount of steady state error correction capability. A default of zero doesn't force the use of RS overhead. Default: 0
phyRRtxRatio	This parameter allows to provision a minimal guaranteed retransmission bandwidth, on top of the minimum rate. In case of the repetitive impulses of known maximal length and periodicity, this parameter can be used to guarantee that the repetitive impulse noise can be corrected. A default of zero doesn't force any extra guaranteed data bandwidth for retransmissions. Default: 0
txPowerAttenuation	Transmit power attenuation. Value ranges from -255 to 255 dB. Default: 0

Table 147: adsl-co-profile parameter definitions (Continued)

Parameter	Description																																
cabMode	ADSL2+ Cabinet Mode Setting Values: Conexant Chipset Based Blades: 0 = OFF 1 - 15 = Sets Cabinet Mode to ON and downstream cut-off frequency is set to 1.1 Mhz. Note: there is no difference for values 1-15 as all values set Cabinet Mode to on. Broadcom Chipset Based Blades: 0 = OFF 1 - 15 = Sets Cabinet Mode to ON and selects TX Filter 1-15. The TX Filters set the downstream cut-off frequency as per the following: <table> <tr> <th>Filter#</th><th>Cutoff Frequency</th></tr> <tr><td>1</td><td>603.75 Khz</td></tr> <tr><td>2</td><td>646.875 Khz</td></tr> <tr><td>3</td><td>690 Khz</td></tr> <tr><td>4</td><td>733.125 Khz</td></tr> <tr><td>5</td><td>776.25 Khz</td></tr> <tr><td>6</td><td>819.375 Khz</td></tr> <tr><td>7</td><td>862.5 Khz</td></tr> <tr><td>8</td><td>905.625 Khz</td></tr> <tr><td>9</td><td>948.75 Khz</td></tr> <tr><td>10</td><td>991.875 Khz</td></tr> <tr><td>11</td><td>1035 Khz</td></tr> <tr><td>12</td><td>1078.125 Khz</td></tr> <tr><td>13</td><td>1121.25 Khz</td></tr> <tr><td>14</td><td>1164.375 Khz</td></tr> <tr><td>15</td><td>1207.5 Khz</td></tr> </table> Default: 0	Filter#	Cutoff Frequency	1	603.75 Khz	2	646.875 Khz	3	690 Khz	4	733.125 Khz	5	776.25 Khz	6	819.375 Khz	7	862.5 Khz	8	905.625 Khz	9	948.75 Khz	10	991.875 Khz	11	1035 Khz	12	1078.125 Khz	13	1121.25 Khz	14	1164.375 Khz	15	1207.5 Khz
Filter#	Cutoff Frequency																																
1	603.75 Khz																																
2	646.875 Khz																																
3	690 Khz																																
4	733.125 Khz																																
5	776.25 Khz																																
6	819.375 Khz																																
7	862.5 Khz																																
8	905.625 Khz																																
9	948.75 Khz																																
10	991.875 Khz																																
11	1035 Khz																																
12	1078.125 Khz																																
13	1121.25 Khz																																
14	1164.375 Khz																																
15	1207.5 Khz																																
ginpAdslCoSupport	Enable or disable downstream G.INP / ITU-G.998.4. Only supported by Broadcom ports. 1 enable 2 disable Default: disable																																
ginpAdslCoEtrMax	Maximum allowed value for downstream expected throughput (ETR) in kbit/s. The valid values are all multiples of 8 from 0 to the maximum of the valid values of the maximum net data rate specified in the associated Recommendation. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 32736 kbps																																

Table 147: adsl-co-profile parameter definitions (Continued)

Parameter	Description
ginpAdslCoEtrMin	Minimum allowed value for downstream expected throughput (ETR) in kbit/s. The valid values are all multiples of 8 from 0 to the maximum of the valid values of the minimum net data rate specified in the associated Recommendation. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 64 kbps
ginpAdslCoNdrMax	Maximum allowed value for downstream net data rate (NDR) in kbit/s. The valid values are all multiples of 8 from 0 to the maximum of the valid values of the maximum net data rate specified in the associated Recommendation. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 32736
ginpAdslCoShineRatio	The downstream loss of rate in a 1 second interval expressed as a fraction of NDR due to a single high impulse noise event (SHINE) impulse noise environment expected by the operator to occur at a probability acceptable for the services. The valid values are all multiples of 0.001 from 0 to 0.1. This field uses 1 to equal 0.001 and 100 to equal 0.1. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 valid configurations. Default: 10
ginpAdslCoLeftrThreshold	The downstream rate Threshold below which the Low Error Free Rate (LEFTR) defect is declared. The threshold used to declare leftr defects is expressed in fraction of the net data rate (NDR). The value 0 is a special value to indicate that the receiver shall use a special value for declaring leftr defect. The minimum valid threshold to declare leftr is ETR/2. The receiver shall ignore threshold values that are less than the minimum and shall use ETR/2 for declaring leftr defect instead. The valid values are all multiples of 0.01 from 0.01 to 0.99. This field uses 1 to equal 0.01 and 99 to equal 0.99. Default: 0
ginpAdslCoMaxDelay	The maximum downstream delay in ms. This is the upper limit for the delay that is added to the transmission delay only caused by retransmissions. Here the receiver anAdding support for G.INP / ITU-T G.998.4d/or the transmitter shall identify and discard all DTUs whose payload cannot be transferred over the reference point at the receiver without violating the delay_max limit. The time stamp shall be the criterion for discarding the DTUs. The processing delay between the U-interface and the retransmission sub-layer of the receiver in the retransmission data path direction shall be excluded from consideration for delay_max in the retransmission data path direction. The valid values are all integers from 1 to 63. ITU-T G.998.4 7.1.1 Control parameters, 7.1.2 Valid configurations, and 81.6 Time Stamp. Default: 20 mSec
ginpAdslCoMinDelay	

Table 147: adsl-co-profile parameter definitions (Continued)

Parameter	Description
ginpAdslCoMin	The minimum downstream impulse noise protection (INP) against single high impulse noise event (SHINE) in discrete multitone (DMT) symbols. The valid values are all integers from 0 to 63 for system with a sub-carrier spacing of 4.3125 kHz. The valid values are all integers from 0 to 127 for system with a sub-carrier spacing of 8.625 kHz. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 4
ginpAdslCoMinRSoverhead	This value specifies the downstream bandwidth reserved for RS (reed-solomon) codewords. The minimum guaranteed R/N ratio. The unit is 1/256th and the range is 0..64 (0 to 25%). Default: 0
ginpAdslCoReinCfg	The minimum downstream impulse protection against electrical repetitive impulse noise (REIN) in DMT symbols. The valid values are all integers from 0 to 7 for system with a sub-carrier spacing of 4.3125 kHz. The valid values are all integers from 0 to 13 for system with a sub-carrier spacing of 8.625 kHz. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations.

Table 147: adsl-co-profile parameter definitions (Continued)

Parameter	Description
ginpAdslCoReinFreq	Specifies the frequency of REIN inter-arrival time. It is used in the Channel Initialization Policy and on-line reconfiguration procedures. REIN is commonly coupled from electrical power cables appliances drawing power from the AC electrical power network, having a repetition rate of twice the AC power frequency (100 or 120 Hz). The valid values are integers 100 hz or 120 hz. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. freq100hz freq120hz Default: freq120hz
ginpAdslCoRtxMode	Downstream retransmission Mode (RTX MODE). The RTX_MODE is a configuration parameter used to control activation of retransmission during initialization. This parameter has 4 valid values: FORBIDDEN: ITU-T G.998.4 retransmission not allowed. PREFERRED: ITU-T G.998.4 retransmission is preferred by the operator. (i.e., if ITU-T G.998.4 RTX capability is supported by both XTU's, the XTU's shall select ITU-T G.998.4 operation for this direction).FORCED: Force the use of the ITU-T G.998.4 retransmission.(i.e., if ITU-T G.998.4 RTX capability in this direction is not supported by both XTU's or not selected by the XTU's, an initialization failure shall result). NOTE: Due to the optionality of ITU-T G.998.4 retransmission in upstream direction, the use of FORCED in upstream may lead to initialization failure, even if the XTU is supporting ITU-T G.998.4 (in downstream). TESTMODE: Force the use of the ITU-T G.998.4 retransmission in the test mode described in clause 10.4. (i.e., if ITU-T G.998.4 RTX capability is not supported by both XTU's or not selected by the XTU's, an initialization failure shall result).ITU-T G.998.4 11.1.13 Retransmission Mode (RTX_MODE). forbidden preferred forced testmode Default: preferred

View adsl-cpe-profile parameter defaults

The ADSL2+ upstream interface is **adsl-cpe-profile** which defines the upstream behavior.

Viewing the adsl-cpe-profile defaults

View the **adsl-cpe-profile** parameters and their default values.

- 1 View the ADSL2+ cards.

```

zSH> slots
MXK 819
Uplinks
  a:*MXK EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK EIGHT GIGE (NOT_PROV)
Cards
  1: MXK ADSL-48-A Bonded (RUNNING)
  2: MXK ADSL-48-A Bonded (RUNNING)
  11: MXK ADSL-48-A Bonded/with 900 Ohm Splitter (RUNNING)
  17: MXK ADSL-48-A Bonded/with 600 Ohm Splitter (RUNNING)

```

2 View the default parameters for the **adsl-cpe-profile**:

```

zSH> get adsl-cpe-profile 1/17/1
adsl-cpe-profile 1/17/1
rateMode: -----> {adaptatruntime}
rateChanRatio: -----> {50}
targetSnrMgn: -----> {60}
maxSnrMgn: -----> {310}
minSnrMgn: -----> {0}
downshiftSnrMgn: -----> {30}
upshiftSnrMgn: -----> {90}
minUpshiftSnrMgn: -----> {60}
minDownshiftSnrMgn: -----> {60}
fastMinTxRate: -----> {32000}
interleaveMinTxRate: -----> {32000}
fastMaxTxRate: -----> {1024000}
interleaveMaxTxRate: -----> {1536000}
maxInterleaveDelay: -----> {16}
thresh15MinLofs: -----> {0}
thresh15MinLoss: -----> {0}
thresh15MinLprs: -----> {0}
thresh15MinESs: -----> {0}
threshFastRateUp: -----> {0}
threshInterleaveRateUp: ---> {0}
threshFastRateDown: -----> {0}
threshInterleaveRateDown: -> {0}
minTxThresholdRateAlarm: --> {0}
minINP: -----> {20}
phyRSupport: -----> {disable}
phyRmaxINP: -----> {0}
phyRminRSoverhead: -----> {0}
phyRRtxRatio: -----> {0}

```

[Table 148](#) defines the values for the **adsl-cpe-profile** parameters

:

Table 148: adsl-cpe-profile parameter definitions

Parameter	Description
rateMode	The transmit rate adaptation configured on this modem. Supported values: fixed: The rate is negotiated at startup and remains fixed. Modem speed is determined by the fastMaxTxRate or interleaveMaxTxRate parameters. adaptatstartup: The rate is negotiated at startup and remains fixed. Modem speed is determined by the fastMaxTxRate or interleaveMaxTxRate parameters. If the line is able to support a higher rate, the rate above the minimum is assigned to the available channel (either fast or interleave). adaptatruntime: The rate is negotiated dynamically and can vary between the maximum and minimum configured rates. If the line conditions change during runtime, the line speed is adjusted. Default: adaptatruntime
rateChanRatio	Configured allocation ratio of excess transmit bandwidth between fast and interleaved channels. Default: 50
targetSnrMgn	Target signal to noise margin (in tenths of dBs). This is the noise margin the modem must achieve with a BER of 10^{-7} or better to successfully complete initialization. Default: 60
maxSnrMgn	Maximum acceptable signal/noise margin (in tenths of dBs). If the noise margin rises above this the modem attempts to reduce its power output to optimize its operation. Default: 310
minSnrMgn	Configured Signal/Noise Margin for rate downshift. If the noise margin falls below this level, the modem should attempt to decrease its transmit rate. Default: 0
downshiftSnrMgn	Configured Signal/Noise Margin for rate downshift. If the noise margin falls below this level, the modem should attempt to decrease its transmit rate. Default: 0
upshiftSnrMgn	Minimum time that the current margin is above upshiftSnrMgn before an upshift occurs. Default: 0
minUpshiftSnrMgn	Minimum time that the current margin is below. DownshiftSnrMgn before a downshift occurs. Default: 0

Table 148: adsl-cpe-profile parameter definitions (Continued)

Parameter	Description
minDownshiftSnrMgn	Configured Signal/Noise Margin for rate downshift. If the noise margin falls below this level, the modem should attempt to decrease it's transmit rate. Default: 0
fastMinTxRate	Minimum transmit rate (in bps) for channels configured for fast transmission mode. For a CPE interface, the range is 32 Kbps to 896 Kbps (512 for G.lite). Default: 32 Kbps
interleaveMinTxRate	Minimum transmit rate (in bps) for channels configured for interleaved transmission mode. For a CPE interface, the range is 32Kbps to 896Kbps (1512Kbps for G.Lite). Default: 32 Kbps
fastMaxTxRate	Maximum transmit rate (in bps) for channels configured for fast transmission mode. For a CPE interface, the range is 32Kbps to 1024Kbps (512Kbps for G.Lite). Default: 1024 Kbps
interleaveMaxTxRate	Maximum transmit rate (in bps) for channels configured for interleaved transmission mode. For a CPE interface, the range is 32 Kbps to 1536 Kbps (512 Kbps for G.lite). Default: 1536 Kbps

Table 148: adsl-cpe-profile parameter definitions (Continued)

Parameter	Description
maxInterleaveDelay	Maximum interleave delay for this channel. Interleave delay applies only to the interleave channel and defines the mapping (relative spacing) between subsequent input bytes at the interleave input and their placement in the bit stream at the interleave output. Larger numbers provide greater separation between consecutive input bytes in the output bit stream allowing for improved impulse noise immunity, but at the expense of payload latency. For video, to maximize protection of downstream signal (where impulse problems occur), minimize round-trip latency by minimizing upstream delay use 1 ms upstream and 16 ms downstream. Values: 0 0.5 ms 1 1 ms 2 2 ms 4 4 ms 8 8 ms 16 16 ms 32 32 ms 63 63 ms Default: 63 ms
thresh15MinLofs	The number of Loss of Frame Seconds encountered by an ADSL interface within any given 15 minutes performance data collection period, which causes the SNMP agent to send an adslAtucPerfLofsThreshTrap. Default: 0
thresh15MinLoss	The number of Loss of Signal Seconds encountered by an ADSL interface within any given 15 minutes performance data collection period, which causes the SNMP agent to send an adslAtucPerfLossThreshTrap. Default: 0
thresh15MinLprs	The number of Loss of Power Seconds encountered by an ADSL interface within any given 15 minutes performance data collection period, which causes the SNMP agent to send an adslAtucPerfLprsThreshTrap. Default: 0
thresh15MinESs	The number of Errored Seconds encountered by an ADSL interface within any given 15 minutes performance data collection period, which causes the SNMP agent to send an adslAtucPerfESsThreshTrap. Default: 0
threshFastRateUp	Applies to 'Fast' channels only. Configured change in rate causing an adslAtucRateChangeTrap. Default: 0

Table 148: adsl-cpe-profile parameter definitions (Continued)

Parameter	Description
threshInterleaveRateUp	For `Interleave' channels only. Configured change in rate causing an adslAtucRateChangeTrap. Default: 0
threshFastRateDown	For `Fast' channels only. Configured change in rate causing an adslAtucRateChangeTrap. Default: 0
threshInterleaveRateDown	For `Interleave' channels only. Configured change in rate causing an adslAtucRateChangeTrap. Default: 0
minTxThresholdRateAlarm	Enables the CO (downstream) transmission rate threshold value. If the rate falls below this value, the device sends a trap and an alarm. Default: 0
minINP (already used in the case of normal interleaving)	This parameter (already used in the case of normal interleaving) defines the minimal guaranteed impulse noise protection, provided that the available data bandwidth allowed for retransmissions is not exceeded. Default: 20
phyRSupport	Enable to turn on Phy-R™ parameters. Disable to turn off Phy-R™ parameters. Default: disable
phyRmaxINP	This parameter defines the maximum number of consecutive retransmissions that may take place and therefore bounds the maximal jitter due to retransmissions. A default value of zero doesn't bound the number of consecutive retransmissions (that will however never exceed maxDelay * 4 symbols). Default: 0
phyRminRSoverhead	This new parameter allows to force a minimum amount of RS overhead. This can be used to guarantee a given amount of steady state error correction capability. A default of zero doesn't force the use of RS overhead. Default: 0
phyRRtxRatio	This parameter allows to provision a minimal guaranteed retransmission bandwidth, on top of the minimum rate. In case of the repetitive impulses of known maximal length and periodicity, this parameter can be used to guarantee that the repetitive impulse noise can be corrected. A default of zero doesn't force any extra guaranteed data bandwidth for retransmissions. Default: 0

Table 148: adsl-cpe-profile parameter definitions (Continued)

Parameter	Description
ginpAdslCpeSupport	Enable or disable upstream G.INP / ITU-G.998.4. enable disable Default: disable
ginpAdslCpeEtrMax	Maximum allowed value for upstream expected throughput (ETR) in kbit/s. The valid values are all multiples of 8 from 0 to the maximum of the valid values of the maximum net data rate specified in the associated Recommendation. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 1536 kbps
ginpAdslCpeEtrMin	Minimum allowed value for upstream expected throughput (ETR) in kbit/s. The valid values are all multiples of 8 from 0 to the maximum of the valid values of the minimum net data rate specified in the associated Recommendation. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 64 kbps
ginpAdslCpeNdrMax	Maximum allowed value for upstream net data rate (NDR) in kbit/s. The valid values are all multiples of 8 from 0 to the maximum of the valid values of the maximum net data rate specified in the associated Recommendation. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 1536 kbps
ginpAdslCpeShineRatio	The upstream loss of rate in a 1 second interval expressed as a fraction of NDR due to a single high impulse noise event (SHINE) impulse noise environment expected by the operator to occur at a probability acceptable for the services. The valid values are all multiples of 0.001 from 0 to 0.1. This field uses 1 to equal 0.001 and 100 to equal 0.1. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 valid configurations. Default: 10
ginpAdslCpeLeftrThreshold	The upstream rate Threshold below which the Low Error Free Rate (LEFTR) defect is declared. The threshold used to declare leftr defects is expressed in fraction of the net data rate (NDR). The value 0 is a special value to indicate that the receiver shall use a special value for declaring leftr defect. The minimum valid threshold to declare leftr is ETR/2. The receiver shall ignore threshold values that are less than the minimum and shall use ETR/2 for declaring leftr defect instead. The valid values are all multiples of 0.01 from 0.01 to 0.99. This field uses 1 to equal 0.01 and 99 to equal 0.99. Default: 0

Table 148: adsl-cpe-profile parameter definitions (Continued)

Parameter	Description
ginpAdslCpeMaxDelay	The maximum upstream delay in ms. This is the upper limit for the delay that is added to the transmission delay only caused by retransmissions. Here the receiver an Adding support for G.INP / ITU-T G.998.4d/or the transmitter shall identify and discard all DTUs whose payload cannot be transferred over the reference point at the receiver without violating the delay_max limit. The time stamp shall be the criterion for discarding the DTUs. The processing delay between the U-interface and the retransmission sub-layer of the receiver in the retransmission data path direction shall be excluded from consideration for delay_max in the retransmission data path direction. The valid values are all integers from 1 to 63. ITU-T G.998.4 7.1.1 Control parameters, 7.1.2 Valid configurations, and 8.1.6 Time Stamp. Default: 20 mSecs
ginpAdslCpeMinDelay	
ginpAdslCpeMin	The minimum upstream impulse noise protection (INP) against single high impulse noise event (SHINE) in discrete multitone (DMT) symbols. The valid values are all integers from 0 to 63 for system with a sub-carrier spacing of 4.3125 kHz. The valid values are all integers from 0 to 127 for system with a sub-carrier spacing of 8.625 kHz. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 4
ginpAdslCpeMinRSoverhead	This value specifies the upstream bandwidth reserved for RS (reed-solomon) codewords. The minimum guaranteed R/N ratio. The unit is 1/256th and the range is 0..64 (0 to 25%). Default: 0
ginpAdslCpeReinCfg	The minimum upstream impulse protection against electrical repetitive impulse noise (REIN) in DMT symbols. The valid values are all integers from 0 to 7 for system with a sub-carrier spacing of 4.3125 kHz. The valid values are all integers from 0 to 13 for system with a sub-carrier spacing of 8.625 kHz. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations. Default: 0

Table 148: adsl-cpe-profile parameter definitions (Continued)

Parameter	Description
ginpAdslCpeReinFreq	Specifies the frequency of REIN inter-arrival time. It is used in the Channel Initialization Policy and on-line reconfiguration procedures. REIN is commonly coupled from electrical power cables appliances drawing power from the AC electrical power network, having a repetition rate of twice the AC power frequency (100 or 120 Hz). The valid values are integers 100 hz or 120 hz. ITU-T G.998.4 7.1.1 Control parameters and 7.1.2 Valid configurations freq100hz freq120hz Default: freq120hz
ginpAdslCpeRtxMode	Upstream retransmission Mode (RTX MODE). The RTX_MODE is a configuration parameter used to control activation of retransmission during initialization. This parameter has 4 valid values: FORBIDDEN: ITU-T G.998.4 retransmission not allowed. PREFERRED: ITU-T G.998.4 retransmission is preferred by the operator. (i.e., if ITU-T G.998.4 RTX capability is supported by both XTU's, the XTU's shall select ITU-T G.998.4 operation for this direction). FORCED: Force the use of the ITU-T G.998.4 retransmission. (i.e., if ITU-T G.998.4 RTX capability in this direction is not supported by both XTU's or not selected by the XTU's, an initialization failure shall result). NOTE: Due to the optionality of ITU-T G.998.4 retransmission in upstream direction, the use of FORCED in upstream may lead to initialization failure, even if the XTU is supporting ITU-T G.998.4 (in downstream). TESTMODE: Force the use of the ITU-T G.998.4 retransmission in the test mode described in clause 10.4. (i.e., if ITU-T G.998.4 RTX capability is not supported by both XTU's or not selected by the XTU's, an initialization failure shall result). forbidden preferred forced testmode Default: preferred

Upstream and downstream tone ranges

The MXK supports setting the active upstream and downstream tone ranges for ADSL2+ modems. Since this is not usually required, understand that changing the range of tones can affect the maximum throughput of the channel as well as providing isolation from certain interference.

The following parameters in the **adsl-profile** specify the range of active tones for the DSL modem:

- **AdslMaxDownstreamToneIndex**
- **AdslMinDownstreamToneIndex**
- **AdslMaxUpstreamToneIndex**
- **AdslMinUpstreamToneIndex**



Note: Changing of any of these parameters causes the modem to retrain.

Configure ADSL2+ profiles for Annex M in fast mode

Enabling Annex M enables more tones to the upstream.



Note: If annex M is disabled, these values should be reset.

Configuring ADSL for Annex M in fastonly mode

Table 149 describes the profiles and parameters and suggested values to enable Annex M in fast mode.

Table 149: Profiles and parameters used to configure ADSL2+ for Annex M in fast mode

Profile	Parameter and value
adsl-profile	adslChannelMode: <i>fastonly</i> adslMinDownstreamToneIndex: 64 adslMaxUpstreamToneIndex: 63 adslAnnexMMode: <i>true</i>
adsl-co-profile (optional)	fastMaxTxRate: 16384000 (16 Mb)
adsl-cpe-profile	fastMaxTxRate: 3072000 (3 Mb)

1 Update the **adsl-profile** for Annex M fast mode:

```

zSH> update adsl-profile 1/9/1
adsl-profile 1/9/1
Please provide the following: [q]uit.
adslLineConfProfile: -----> {0000000961}: ** read-only **
adslAlarmConfProfile: -----> {0000000961}: ** read-only **
adslTrellisModeEnabled: -----> {true}:
adslNTRModeEnabled: -----> {false}:
adslTransmissionMode: -----> {autonegotiatemode}:
adslChannelMode: -----> {interleavedonly}: fastonly
adslMaxDownstreamToneIndex: -----> {511}:
adslMinDownstreamToneIndex: -----> {32}: 64
adslMaxUpstreamToneIndex: -----> {31}: 63
adslMinUpstreamToneIndex: -----> {6}:
adslPotsBypassRelayMaxDuration: -> {60}:
adslLineDMTConfMode: -----> {freqdivmux}:

```

```

adslAnnexMModeEnabled: -----> {false}: true
adslAnnexMPsdMask: -----> {eu56}:
adslAnnexJModeEnabled: -----> {false}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

2 Update the **adsl-co-profile** downstream interface and set the **fastMaxTxRate** to 16 Mb. (Optional)

```

zSH> update adsl-co-profile 1/9/1
adsl-co-profile 1/9/1
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minUpshiftTime: -----> {60}:
minDownshiftTime: -----> {60}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {32736000}: 16384000
maxInterleaveDelay: -----> {63}:
interleaveMaxTxRate: -----> {32736000}:
thresh15MinLofs: -----> {0}:
thresh15MinLoss: -----> {0}:
thresh15MinLols: -----> {0}:
thresh15MinLprs: -----> {0}:
thresh15MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: ---> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: -> {0}:
initFailureTrapEnable: ----> {disable}:
reachExtendedAdsl2: -----> {enable}:
minTxThresholdRateAlarm: --> {0}:
minINP: -----> {20}:
phyRSupport: -----> {disable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
txPowerAttenuation: -----> {20}:
cabMode: -----> {0}:
ginpAdslCoSupport: -----> {disable}:
ginpAdslCoEtrMax: -----> {32736}:
ginpAdslCoEtrMin: -----> {64}:
ginpAdslCoNdrMax: -----> {32736}:
ginpAdslCoShineRatio: -----> {10}:
ginpAdslCoLeftrThreshold: -> {0}:
ginpAdslCoMaxDelay: -----> {20}:
ginpAdslCoMinDelay: -----> {0}:

```



```

ginpAdslCoMin: -----> {4}:
ginpAdslCoMinRSoverhead: --> {0}:
ginpAdslCoReinCfg: -----> {0}:
ginpAdslCoReinFreq: -----> {freq120hz}:
ginpAdslCoRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

3 Update the **adsl-cpe-profile** upstream interface and set the **fastMaxTxRate** to 3 Mb.

```

zSH> update adsl-cpe-profile 1/9/1
adsl-cpe-profile 1/9/1
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minUpshiftSnrMgn: -----> {60}:
minDownshiftSnrMgn: -----> {60}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {1024000}: 3072000
interleaveMaxTxRate: -----> {1536000}:
maxInterleaveDelay: -----> {16}:
thresh15MinLofs: -----> {0}:
thresh15MinLoss: -----> {0}:
thresh15MinLprs: -----> {0}:
thresh15MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: ----> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: --> {0}:
minTxThresholdRateAlarm: ---> {0}:
minINP: -----> {20}:
phyRSupport: -----> {disable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
ginpAdslCpeSupport: -----> {disable}:
ginpAdslCpeEtrMax: -----> {1536}:
ginpAdslCpeEtrMin: -----> {64}:
ginpAdslCpeNdrMax: -----> {1536}:
ginpAdslCpeShineRatio: -----> {10}:
ginpAdslCpeLeftrThreshold: -> {0}:
ginpAdslCpeMaxDelay: -----> {20}:
ginpAdslCpeMinDelay: -----> {0}:
ginpAdslCpeMin: -----> {4}:
ginpAdslCpeMinRSoverhead: --> {0}:
ginpAdslCpeReinCfg: -----> {0}:

```

```
ginpAdslCpeReinFreq: -----> {freq120hz}:
ginpAdslCpeRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Configure ADSL2+ profiles for Annex M in interleaved mode

Enabling Annex M enables more tones to the upstream.



Note: If Annex M is disabled, these values should be reset.

Configuring ADSL2+ for Annex M in interleaved mode

Table 150 describes the profiles and parameters and suggested values to enable Annex M in interleave mode.

Table 150: Profiles and parameters used to configure ADSL2+ for Annex M interleave mode

Profile	Parameter and value
adsl-profile	adslChannelMode: <i>interleavedonly</i> adslMinDownstreamToneIndex: 64 adslMaxUpstreamToneIndex: 63 adslAnnexMMode: <i>true</i>
adsl-co-profile (optional)	interleaveMaxTxRate: 16384000 (16 Mb)
adsl-cpe-profile	interleaveMaxTxRate: 3072000 (3 Mb) minINP: 10

- 1 Update the **adsl-profile** for Annex M interleaved mode.

```
zSH> update adsl-profile 1/9/2
adsl-profile 1/9/2
Please provide the following: [q]uit.
adslLineConfProfile: -----> {0000000961}:** read-only **
adslAlarmConfProfile: -----> {0000000961}:** read-only **
adslTrellisModeEnabled: -----> {true}:
adslNTRModeEnabled: -----> {false}:
adslTransmissionMode: -----> {autonegotiatemode}:
adslChannelMode: -----> {fastonly}: interleavedonly
adslMaxDownstreamToneIndex: -----> {511}:
adslMinDownstreamToneIndex: -----> {32}: 64
adslMaxUpstreamToneIndex: -----> {31}: 63
adslMinUpstreamToneIndex: -----> {6}:
adslPotsBypassRelayMaxDuration: -> {60}:
adslLineDMTConfMode: -----> {freqdivmux}:
adslAnnexMModeEnabled: -----> {false}: true
adslAnnexMPsdMask: -----> {eu56}:
```

```
adslAnnexJModeEnabled: -----> {false}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

2 Update the **adsl-co-profile** downstream interface and set the **interleaveMaxTxRate** to 16 Mb. (Optional)

```
zSH> update adsl-co-profile 1/9/2
adsl-co-profile 1/9/2
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minUpshiftTime: -----> {60}:
minDownshiftTime: -----> {60}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {32736000}: 16384000
maxInterleaveDelay: -----> {63}:
interleaveMaxTxRate: -----> {32736000}:
thresh15MinLofs: -----> {0}:
thresh15MinLoss: -----> {0}:
thresh15MinLols: -----> {0}:
thresh15MinLprs: -----> {0}:
thresh15MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: ---> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: -> {0}:
initFailureTrapEnable: ----> {disable}:
reachExtendedAdsl2: -----> {enable}:
minTxThresholdRateAlarm: --> {0}:
minINP: -----> {20}:
phyRSupport: -----> {disable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
txPowerAttenuation: -----> {20}:
cabMode: -----> {0}:
ginpAdslCoSupport: -----> {disable}:
ginpAdslCoEtrMax: -----> {32736}:
ginpAdslCoEtrMin: -----> {64}:
ginpAdslCoNdrMax: -----> {32736}:
ginpAdslCoShineRatio: -----> {10}:
ginpAdslCoLeftrThreshold: -> {0}:
ginpAdslCoMaxDelay: -----> {20}:
ginpAdslCoMinDelay: -----> {0}:
ginpAdslCoMin: -----> {4}:
ginpAdslCoMinRSoverhead: --> {0}:
```

```

ginpAdslCoReinCfg: -----> {0}:
ginpAdslCoReinFreq: -----> {freq120hz}:
ginpAdslCoRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

3 Update the **adsl-cpe-profile** upstream interface and set the **interleaveMaxTxRate** to 3 Mb and set the **minINP** to 10.



Note: When configuring the **adsl-cpe-profile** for Annex M interleaved, the default setting for the **minINP** parameter must be set to less than two symbols (a value under 20) in order to achieve optimal upstream train rates.

Zhone recommends setting the **minINP** parameter to **10** to achieve a balance between protection against RFI and protection from impulse noise.

```

zSH> update adsl-cpe-profile 1/9/2
adsl-cpe-profile 1/9/2
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minUpshiftSnrMgn: -----> {60}:
minDownshiftSnrMgn: -----> {60}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {1024000}:
interleaveMaxTxRate: -----> {1536000}: 3072000
maxInterleaveDelay: -----> {16}:
thresh15MinLofs: -----> {0}:
thresh15MinLoss: -----> {0}:
thresh15MinLprs: -----> {0}:
thresh15MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: -----> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: -----> {0}:
minTxThresholdRateAlarm: -----> {0}:
minINP: -----> {20}: 10
phyRSupport: -----> {disable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
ginpAdslCpeSupport: -----> {disable}:
ginpAdslCpeEtrMax: -----> {1536}:
ginpAdslCpeEtrMin: -----> {64}:

```

```

ginpAdslCpeNdrMax: -----> {1536}:
ginpAdslCpeShineRatio: -----> {10}:
ginpAdslCpeLeftrThreshold: -> {0}:
ginpAdslCpeMaxDelay: -----> {20}:
ginpAdslCpeMinDelay: -----> {0}:
ginpAdslCpeMin: -----> {4}:
ginpAdslCpeMinRSoverhead: --> {0}:
ginpAdslCpeReinCfg: -----> {0}:
ginpAdslCpeReinFreq: -----> {freq120hz}:
ginpAdslCpeRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Configure ADSL2+ profiles for G.lite

Configuring ADSL 2+ profiles for G.lite

When the transmission mode is set to G.lite, the channel mode must be set to interleaved.

[Table 151](#) describes the profiles and parameters and suggested values to enable G.lite mode.

Table 151: Profile and parameters used to configure ADSL2+ for G.lite

Profile	Parameter and value
adsl-profile	adslTransmissionMode: <i>glitemode</i> adslChannelMode: <i>interleavedonly</i>
adsl-co-profile	interleaveMaxTxRate: <i>1536000 kbps</i>
adsl-cpe-profile	interleaveMaxTxRate: <i>512000 kbps</i>

- 1 If you change the transmission rate to *glitemode* and the channel mode to *interleavedonly* in the **adsl-profile** and save the change, you may see this error message:

```

.....
Save changes? [s]ave, [c]hange or [q]uit: s
interleaveMaxTxRate set too high in ADSL CO and CPE profiles to select
glitemode.

```

In this case, configure the **interleaveMaxTxRate** parameter in the **adsl-co-profile** and the **adsl-cpe-profile** to a correct value, then configure the **adsl-profile** for G.lite.

A valid value for the **adsl-co-profile interleaveMaxTxRate** is 1536 Kbps or less.

Set the **interleaveMaxTxRate** in the **adsl-co-profile**:

```
zSH> update adsl-co-profile 1/9/5
```

```

adsl-co-profile 1/9/5
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minUpshiftTime: -----> {60}:
minDownshiftTime: -----> {60}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {32736000}:
maxInterleaveDelay: -----> {63}:
interleaveMaxTxRate: -----> {32736000}: 1536000
thresh15MinLofs: -----> {0}:
thresh15MinLoss: -----> {0}:
thresh15MinLols: -----> {0}:
thresh15MinLprs: -----> {0}:
thresh15MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: ---> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: -> {0}:
initFailureTrapEnable: ----> {disable}:
reachExtendedAdsl2: -----> {enable}:
minTxThresholdRateAlarm: --> {0}:
minINP: -----> {20}:
phyRSupport: -----> {disable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
txPowerAttenuation: -----> {20}:
cabMode: -----> {0}:
ginpAdslCoSupport: -----> {disable}:
ginpAdslCoEtrMax: -----> {32736}:
ginpAdslCoEtrMin: -----> {64}:
ginpAdslCoNdrMax: -----> {32736}:
ginpAdslCoShineRatio: -----> {10}:
ginpAdslCoLeftrThreshold: -> {0}:
ginpAdslCoMaxDelay: -----> {20}:
ginpAdslCoMinDelay: -----> {0}:
ginpAdslCoMin: -----> {4}:
ginpAdslCoMinRSoverhead: --> {0}:
ginpAdslCoReinCfg: -----> {0}:
ginpAdslCoReinFreq: -----> {freq120hz}:
ginpAdslCoRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

- 2** A valid value for the **adsl-cpe-profile interleaveMaxTxRate** is 512 Kbps or less.

Set the **interleaveMaxTxRate** in the **adsl-cpe-profile**

```

zSH> update adsl-cpe-profile 1/9/5
adsl-cpe-profile 1/9/5
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minUpshiftSnrMgn: -----> {60}:
minDownshiftSnrMgn: -----> {60}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {1024000}:
interleaveMaxTxRate: -----> {1536000}: 512000
maxInterleaveDelay: -----> {16}:
threshl5MinLofs: -----> {0}:
threshl5MinLoss: -----> {0}:
threshl5MinLprs: -----> {0}:
threshl5MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: ----> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: --> {0}:
minTxThresholdRateAlarm: ---> {0}:
minINP: -----> {20}:
phyRSupport: -----> {disable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
ginpAdslCpeSupport: -----> {disable}:
ginpAdslCpeEtrMax: -----> {1536}:
ginpAdslCpeEtrMin: -----> {64}:
ginpAdslCpeNdrMax: -----> {1536}:
ginpAdslCpeShineRatio: -----> {10}:
ginpAdslCpeLefttrThreshold: -> {0}:
ginpAdslCpeMaxDelay: -----> {20}:
ginpAdslCpeMinDelay: -----> {0}:
ginpAdslCpeMin: -----> {4}:
ginpAdslCpeMinRSoverhead: --> {0}:
ginpAdslCpeReinCfg: -----> {0}:
ginpAdslCpeReinFreq: -----> {freq120hz}:
ginpAdslCpeRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

3 Update the **adsl-profile** for G.lite:

```

zSH> update adsl-profile 1/9/5
adsl-profile 1/9/5

```

```
Please provide the following: [q]uit.
adslLineConfProfile: -----> {0000000961}: ** read-only **
adslAlarmConfProfile: -----> {0000000961}: ** read-only **
adslTrellisModeEnabled: -----> {true}:
adslNTRModeEnabled: -----> {false}:
adslTransmissionMode: -----> {autonegotiatemode}: glitemode
adslChannelMode: -----> {fastonly}: interleavedonly
adslMaxDownstreamToneIndex: -----> {511}:
adslMinDownstreamToneIndex: -----> {32}:
adslMaxUpstreamToneIndex: -----> {31}:
adslMinUpstreamToneIndex: -----> {6}:
adslPotsBypassRelayMaxDuration: -> {60}:
adslLineDMTConfMode: -----> {freqdivmux}:
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu56}:
adslAnnexJModeEnabled: -----> {false}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Configure ADSL2+ profiles to cap train rates

This section provides typical examples of capping upstream and downstream train rates for both fast and interleaved mode.

[Table 152](#) describes the profiles and parameters and suggested upstream and downstream train rates.

Table 152: Profiles and parameters for capping upstream and downstream train rates

Profile	Parameter and train rates
adsl-profile	adslChannelMode: <i>fastonly</i> or <i>interleavedonly</i>
adsl-co-profile	fastMaxTxRate: 20,000,000 bps or interleaveMaxTxRate: 20,000,000 bps
adsl-cpe-profile	Note: bps show some typical train rates for the upstream. fastTaxTxRate: 384,000 bps or interleaveMaxTxRate: 512,000 bps

Capping upstream and downstream train rates (fast mode)

- 1 Change the channel mode in the **adsl-profile** to *fastonly*:

```
zSH> update adsl-profile 1/9/10
adsl-profile 1/9/10
Please provide the following: [q]uit.
adslLineConfProfile: -----> {0000000961}: ** read-only **
adslAlarmConfProfile: -----> {0000000961}: ** read-only **
```



```

adslTrellisModeEnabled: -----> {true}:
adslNTRModeEnabled: -----> {false}:
adslTransmissionMode: -----> {autonegotiatemode}:
adslChannelMode: -----> {interleavedonly}: fastonly
adslMaxDownstreamToneIndex: -----> {511}:
adslMinDownstreamToneIndex: -----> {32}:
adslMaxUpstreamToneIndex: -----> {31}:
adslMinUpstreamToneIndex: -----> {6}:
adslPotsBypassRelayMaxDuration: -> {60}:
adslLineDMTConfMode: -----> {freqdivmux}:
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu56}:
adslAnnexJModeEnabled: -----> {false}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

2 Cap the downstream rate:

```

zSH> update adsl-co-profile 1/9/10
adsl-co-profile 1/9/10
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minUpshiftTime: -----> {60}:
minDownshiftTime: -----> {60}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {32736000}: 20000000
maxInterleaveDelay: -----> {63}:
interleaveMaxTxRate: -----> {32736000}:
thresh15MinLofs: -----> {0}:
thresh15MinLoss: -----> {0}:
thresh15MinLols: -----> {0}:
thresh15MinLprs: -----> {0}:
thresh15MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: ---> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: -> {0}:
initFailureTrapEnable: ----> {disable}:
reachExtendedAdsl2: -----> {enable}:
minTxThresholdRateAlarm: --> {0}:
minINP: -----> {20}:
phyRSupport: -----> {disable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
txPowerAttenuation: -----> {20}:

```

```

cabMode: -----> {0}:
ginpAdslCoSupport: -----> {disable}:
ginpAdslCoEtrMax: -----> {32736}:
ginpAdslCoEtrMin: -----> {64}:
ginpAdslCoNdrMax: -----> {32736}:
ginpAdslCoShineRatio: -----> {10}:
ginpAdslCoLeftrThreshold: --> {0}:
ginpAdslCoMaxDelay: -----> {20}:
ginpAdslCoMinDelay: -----> {0}:
ginpAdslCoMin: -----> {4}:
ginpAdslCoMinRSoverhead: --> {0}:
ginpAdslCoReinCfg: -----> {0}:
ginpAdslCoReinFreq: -----> {freq120hz}:
ginpAdslCoRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

3 Cap the upstream rate:

```

zSH> update adsl-cpe-profile 1/9/10
adsl-cpe-profile 1/9/10
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minUpshiftSnrMgn: -----> {60}:
minDownshiftSnrMgn: -----> {60}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {1024000}: 384000
interleaveMaxTxRate: -----> {1536000}:
maxInterleaveDelay: -----> {16}:
threshl5MinLofs: -----> {0}:
threshl5MinLoss: -----> {0}:
threshl5MinLprs: -----> {0}:
threshl5MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: ----> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: --> {0}:
minTxThresholdRateAlarm: ---> {0}:
minINP: -----> {20}:
phyRSupport: -----> {disable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
ginpAdslCpeSupport: -----> {disable}:
ginpAdslCpeEtrMax: -----> {1536}:
ginpAdslCpeEtrMin: -----> {64}:

```

```

ginpAdslCpeNdrMax: -----> {1536}:
ginpAdslCpeShineRatio: -----> {10}:
ginpAdslCpeLeftrThreshold: -> {0}:
ginpAdslCpeMaxDelay: -----> {20}:
ginpAdslCpeMinDelay: -----> {0}:
ginpAdslCpeMin: -----> {4}:
ginpAdslCpeMinRSoverhead: --> {0}:
ginpAdslCpeReinCfg: -----> {0}:
ginpAdslCpeReinFreq: -----> {freq120hz}:
ginpAdslCpeRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Capping upstream and downstream train rates (interleaved mode)

1 Change the channel mode in the **adsl-profile** to *interleavedonly*:

```

zSH> update adsl-profile 1/9/11
adsl-profile 1/9/11
Please provide the following: [q]uit.
adslLineConfProfile: -----> {0000000961}: ** read-only **
adslAlarmConfProfile: -----> {0000000961}: ** read-only **
adslTrellisModeEnabled: -----> {true}:
adslNTRModeEnabled: -----> {false}:
adslTransmissionMode: -----> {autonegotiatemode}:
adslChannelMode: -----> {fastonly}: interleavedonly
adslMaxDownstreamToneIndex: -----> {511}:
adslMinDownstreamToneIndex: -----> {32}:
adslMaxUpstreamToneIndex: -----> {31}:
adslMinUpstreamToneIndex: -----> {6}:
adslPotsBypassRelayMaxDuration: -> {60}:
adslLineDMTConfMode: -----> {freqdivmux}:
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu56}:
adslAnnexJModeEnabled: -----> {false}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

2 Cap the downstream rate:

```

zSH> update adsl-co-profile 1/9/11
adsl-co-profile 1/9/11
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minUpshiftTime: -----> {60}:

```

```

minDownshiftTime: -----> {60}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {32736000}: 20000000
maxInterleaveDelay: -----> {63}:
interleaveMaxTxRate: -----> {32736000}:
thresh15MinLofs: -----> {0}:
thresh15MinLoss: -----> {0}:
thresh15MinLols: -----> {0}:
thresh15MinLprs: -----> {0}:
thresh15MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: ---> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: -> {0}:
initFailureTrapEnable: ----> {disable}:
reachExtendedAdsl2: -----> {enable}:
minTxThresholdRateAlarm: --> {0}:
minINP: -----> {20}:
phyRSupport: -----> {disable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
txPowerAttenuation: -----> {20}:
cabMode: -----> {0}:
ginpAdslCoSupport: -----> {disable}:
ginpAdslCoEtrMax: -----> {32736}:
ginpAdslCoEtrMin: -----> {64}:
ginpAdslCoNdrMax: -----> {32736}:
ginpAdslCoShineRatio: -----> {10}:
ginpAdslCoLeftrThreshold: -> {0}:
ginpAdslCoMaxDelay: -----> {20}:
ginpAdslCoMinDelay: -----> {0}:
ginpAdslCoMin: -----> {4}:
ginpAdslCoMinRSoverhead: --> {0}:
ginpAdslCoReinCfg: -----> {0}:
ginpAdslCoReinFreq: -----> {freq120hz}:
ginpAdslCoRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

3 Cap the upstream rate:

```

zSH> update adsl-cpe-profile 1/9/11
adsl-cpe-profile 1/9/11
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:

```

```

minUpshiftSnrMgn: -----> {60}:
minDownshiftSnrMgn: -----> {60}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {1024000}:
interleaveMaxTxRate: -----> {1536000}:512000 bps
maxInterleaveDelay: -----> {16}:
thresh15MinLofs: -----> {0}:
thresh15MinLoss: -----> {0}:
thresh15MinLprs: -----> {0}:
thresh15MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: ----> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: --> {0}:
minTxThresholdRateAlarm: ---> {0}:
minINP: -----> {20}:
phyRSupport: -----> {disable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
ginpAdslCpeSupport: -----> {disable}:
ginpAdslCpeEtrMax: -----> {1536}:
ginpAdslCpeEtrMin: -----> {64}:
ginpAdslCpeNdrMax: -----> {1536}:
ginpAdslCpeShineRatio: -----> {10}:
ginpAdslCpeLeftrThreshold: -> {0}:
ginpAdslCpeMaxDelay: -----> {20}:
ginpAdslCpeMinDelay: -----> {0}:
ginpAdslCpeMin: -----> {4}:
ginpAdslCpeMinRSoverhead: --> {0}:
ginpAdslCpeReinCfg: -----> {0}:
ginpAdslCpeReinFreq: -----> {freq120hz}:
ginpAdslCpeRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Configure ADSL2+ S=1/2

This section describes S=1/2 mode transmission on ADSL2+ line cards.

The ADSL2+ S=1/2 specification, as defined in the ITU standard G.992.2, is a transmission mode that supports downstream data rates up to 12 Mbps at distances of 6,000 feet or less. ADSL2+=1/2 is configured in either fast mode or interleaved mode. See [Configuring S=1/2 transmission mode for fast mode on page 1251](#) and [Configuring S=1/2 transmission mode for interleaved mode on page 1253](#).

Configure interleaved channels in the **adsl-profile**:

Table 153: adsl-profile

adsl-profile	Description Both ATU-C and ATU-R	MXK support	Range supported	Default
adslLineConfProfile	Read-Only	PARTIAL	260 only	260
adslAlarmConfProfile	Read-Only	PARTIAL	261 only	260
adslTrellisModeEnabled	Enable/Disable Trellis Mode	Yes	Enable=TRUE/ Disable=FALSE	TRUE
adslINTRModeEnabled	Network Timing Recovery	Not Used	Enable=TRUE/ Disable=FALSE	TRUE
adslTransmissionMode	Sets Transmission Mode	Yes	Yes	Yes
adslChannelMode	Specifies Channelization (Fast/Interleave)	Per line (not per channel)	fastonly interleavedonly	fastonly
adslMaxDownstreamToneIndex	Maximum Downstream Active Tones	Yes	6 to 1023	511
adslMinDownstreamToneIndex	Minimum Downstream Active Tones	Yes	6 to 1023	32
adslMaxUpstreamToneIndex	Maximum Upstream Active Tones	Yes	1 to 63	31
adslMinUpstreamToneIndex	Minimum Upstream Active Tones	Yes	1 to 63	6
adslPotsBypassRelayMaxDuration	Not Used	Not Used	1 to 300	Not Used
adslLineDMTConfMode	DMT Mode - Echo Cancel or Freq Div Mux	Freq Div Mux Only	Freq Div Mux Only	Freq Div Mux
adslAnnexMModeEnabled	Enable/Disable Annex-M	Yes	Enable=TRUE/ Disable=FALSE	FALSE
adslAnnexMPsdMask				

Set the maximum transmit rate in the **adsl-co-profile**:

Table 154: adsl-co-profile

adsl-co-profile	ATU-C	SLMS Supported	SLMS Range supported	SLMS Default
rateMode	Transmit Rate Adaptation	Yes	AdaptAtRuntime Only	AdaptAtRuntime
rateChanRatio	Ratio of avail versus min rates	Defaulted	0 to 100	Defaulted
targetSnrMgn	Target Signal to Noise Ratio (SNR)	Yes	Yes	Yes

Table 154: adsl-co-profile (Continued)

adsl-co-profile	ATU-C	SLMS Supported	SLMS Range supported	SLMS Default
maxSnrMgn	Maximum SNR	Yes	Yes	Yes
minSnrMgn	Minimum SNR	Yes	Yes	Yes
downshiftSnrMgn	Seamless Rate Adaptation	no	NA	no
upshiftSnrMgn	Seamless Rate Adaptation	no	NA	no
minUpshiftTime	Seamless Rate Adaptation	no	NA	no
minDownshiftTime	Seamless Rate Adaptation	no	NA	no
fastMinTxRate	Minimum Transmit Rate for channels configured as Fast	Yes	Yes	Yes
interleaveMinTxRate	Minimum Transmit Rate for channels configured as Interleaved	Yes	Yes	Yes
fastMaxTxRate	Maximum Transmit Rate for channels configured as Fast	Yes	Yes	Yes
maxInterleaveDelay	Maximum Interleave Delay for channel(s) configured as Interleaved	Yes	1 to 63	63 when in ADSL2+ Annex A
interleaveMaxTxRate	Maximum Transmit Rate for channels configured as Interleaved	Yes	Yes	Yes
thresh15MinLofs	Loss of Frame event count	Yes	Yes	Yes
thresh15MinLoss	Loss of signal event count	Yes	Yes	Yes
thresh15MinLols	Loss of link event count	Yes	Yes	Yes
thresh15MinLprs	Loss of Loss of Power Seconds event count	Defaulted	Defaulted	Defaulted
thresh15MinESs	Errored Seconds event count	Yes	Yes	Yes
threshFastRateUp	Threshold time for increase rate on channels configured as Fast	Defaulted	Defaulted	Defaulted

Configuring S=1/2 transmission mode for fast mode

- 1 Verify that the **adminstatus** of the ADSL2+ port is *up*:

```
zSH> port up 1-17-1-0/adsl
1-17-1-0/adsl set to admin state UP
```

2 Verify that the ADSL2+ channelization is set to *fastonly*:

```

zSH> update adsl-profile 1/17/1
adsl-profile 1/17/1
Please provide the following: [q]uit.
adslLineConfProfile: -----> {0000000961}: ** read-only **
adslAlarmConfProfile: -----> {0000000961}: ** read-only **
adslTrellisModeEnabled: -----> {true}:
adslNTRModeEnabled: -----> {false}:
adslTransmissionMode: -----> {autonegotiatemode}:
adslChannelMode: -----> {interleaveonly}: fastonly
adslMaxDownstreamToneIndex: -----> {511}:
adslMinDownstreamToneIndex: -----> {32}:
adslMaxUpstreamToneIndex: -----> {31}:
adslMinUpstreamToneIndex: -----> {6}:
adslPotsBypassRelayMaxDuration: -> {60}:
adslLineDMTConfMode: -----> {freqdivmux}:
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu56}:
adslAnnexJModeEnabled: -----> {false}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

3 Set the maximum transmit rate to 12 Mbps for fast ADSL2+ channel modes. This forces the ADSL2+ port into S=1/2 transmission mode.

```

zSH> update adsl-co-profile 1/17/1
adsl-co-profile 1/17/1
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minUpshiftTime: -----> {60}:
minDownshiftTime: -----> {60}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {32736000}: 12000000
maxInterleaveDelay: -----> {63}:
interleaveMaxTxRate: -----> {32736000}:
thresh15MinLofs: -----> {0}:
thresh15MinLoss: -----> {0}:
thresh15MinLols: -----> {0}:
thresh15MinLprs: -----> {0}:
thresh15MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: ---> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: -> {0}:
initFailureTrapEnable: ----> {disable}:

```



```

reachExtendedAdsl2: -----> {enable}:
minTxThresholdRateAlarm: --> {0}:
minINP: -----> {20}:
phyRSupport: -----> {disable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
txPowerAttenuation: -----> {20}:
cabMode: -----> {0}:
ginpAdslCoSupport: -----> {disable}:
ginpAdslCoEtrMax: -----> {32736}:
ginpAdslCoEtrMin: -----> {64}:
ginpAdslCoNdrMax: -----> {32736}:
ginpAdslCoShineRatio: -----> {10}:
ginpAdslCoLeftrThreshold: -> {0}:
ginpAdslCoMaxDelay: -----> {20}:
ginpAdslCoMinDelay: -----> {0}:
ginpAdslCoMin: -----> {4}:
ginpAdslCoMinRSoverhead: --> {0}:
ginpAdslCoReinCfg: -----> {0}:
ginpAdslCoReinFreq: -----> {freq120hz}:
ginpAdslCoRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Configuring S=1/2 transmission mode for interleaved mode

1 Verify that the **adminstatus** of the ADSL2+ port is *up*:

```

zSH> port up 1-17-1-0/adsl
1-17-1-0/adsl set to admin state UP

```

2 Set the ADSL2+ channelization to **interleavedonly**:

```

zSH> update adsl-profile 1/17/1
adsl-profile 1/17/1
Please provide the following: [q]uit.
adslLineConfProfile: -----> {0000000961}: ** read-only **
adslAlarmConfProfile: -----> {0000000961}: ** read-only **
adslTrellisModeEnabled: -----> {true}:
adslINTRModeEnabled: -----> {false}:
adslTransmissionMode: -----> {autonegotiatemode}:
adslChannelMode: -----> {fastonly}: interleavedonly
adslMaxDownstreamToneIndex: -----> {511}:
adslMinDownstreamToneIndex: -----> {32}:
adslMaxUpstreamToneIndex: -----> {31}:
adslMinUpstreamToneIndex: -----> {6}:
adslPotsBypassRelayMaxDuration: -> {60}:
adslLineDMTConfMode: -----> {freqdivmux}:
adslAnnexMModeEnabled: -----> {false}:
adslAnnexMPsdMask: -----> {eu56}:
adslAnnexJModeEnabled: -----> {false}:
.....

```

Save changes? [s]ave, [c]hange or [q]uit: **s**
Record updated.

- 3** Set the maximum transmit rate to 12 Mbps for interleaved ADSL2+ channel mode. This forces the ADSL2+ port into S=1/2 transmission mode.

```
zSH> update adsl-co-profile 1/17/1
adsl-co-profile 1/17/1
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {0}:
upshiftSnrMgn: -----> {90}:
minUpshiftTime: -----> {0}:
minDownshiftTime: -----> {0}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {32736000}:
maxInterleaveDelay: -----> {63}:
interleaveMaxTxRate: -----> {32736000}: 12000000
thresh15MinLofs: -----> {0}:
thresh15MinLoss: -----> {0}:
thresh15MinLols: -----> {0}:
thresh15MinLprs: -----> {0}:
thresh15MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: ----> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: -> {0}:
initFailureTrapEnable: ----> {disable}:
reachExtendedAdsl2: -----> {enable}:
minTxThresholdRateAlarm: --> {0}:
minINP: -----> {0}:
phyRSupport: -----> {disable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
txPowerAttenuation: -----> {20}:
cabMode: -----> {0}
ginpAdslCoSupport: -----> {disable}:
ginpAdslCoEtrMax: -----> {32736}:
ginpAdslCoEtrMin: -----> {64}:
ginpAdslCoNdrMax: -----> {32736}:
ginpAdslCoShineRatio: -----> {10}:
ginpAdslCoLeftrThreshold: -> {0}:
ginpAdslCoMaxDelay: -----> {20}:
ginpAdslCoMinDelay: -----> {0}:
ginpAdslCoMin: -----> {4}:
ginpAdslCoMinRSoverhead: --> {0}:
ginpAdslCoReinCfg: -----> {0}:
```

```

ginpAdslCoReinFreq: -----> {freq120hz}:
ginpAdslCoRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Configure Broadcom Phy-R™ parameters

Setting the Broadcom Phy-R™ parameters in the CO and CPE ADSL2+ profiles is for advanced users.



Note: The Phy-R™ parameter in the ADSL2+ CO profile cannot be used unless there is a Broadcom CPE modem at the customer site with Phy-R™ parameters in the ADSL2+ CPE profile.

Table 155 describes the profiles and parameters and suggested values to enable Phy-R™.

Table 155: Profiles and parameters used to configure ADSL2+ for Phy-R™

Parameter	Definition
adsl-co-profile	maxInterleaveDelay: 4 minINP: 20 phyRSupport: enable
adsl-cpe-profile	maxInterleaveDelay: 4 minINP: 20 phyRSupport: enable

Enabling Phy-R™ parameters

Update the **adsl-co-profile** and the **adsl-cpe-profile**:

```

zSH> update adsl-co-profile 1/3/2
adsl-co-profile 1/3/2
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {0}:
upshiftSnrMgn: -----> {0}:
minUpshiftTime: -----> {0}:
minDownshiftTime: -----> {0}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {32736000}:
maxInterleaveDelay: -----> {8}: 4
interleaveMaxTxRate: -----> {32736000}:

```

```

thresh15MinLofs: -----> {0}:
thresh15MinLoss: -----> {0}:
thresh15MinLols: -----> {0}:
thresh15MinLprs: -----> {0}:
thresh15MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: ---> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: -> {0}:
initFailureTrapEnable: ----> {disable}:
reachExtendedAdsl2: -----> {enable}:
minTxThresholdRateAlarm: --> {0}:
minINP: -----> {0}: 20
phyRSupport: -----> {disable}: enable
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
txPowerAttenuation: ----> {20}:
cabMode: -----> {0}
ginpAdslCoSupport: -----> {disable}:
ginpAdslCoEtrMax: -----> {32736}:
ginpAdslCoEtrMin: -----> {64}:
ginpAdslCoNdrMax: -----> {32736}:
ginpAdslCoShineRatio: -----> {10}:
ginpAdslCoLeftrThreshold: -> {0}:
ginpAdslCoMaxDelay: -----> {20}:
ginpAdslCoMinDelay: -----> {0}:
ginpAdslCoMin: -----> {4}:
ginpAdslCoMinRSoverhead: --> {0}:
ginpAdslCoReinCfg: -----> {0}:
ginpAdslCoReinFreq: -----> {freq120hz}:
ginpAdslCoRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

```

zSH> update adsl-cpe-profile 1/3/2
adsl-cpe-profile 1/3/2
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minUpshiftSnrMgn: -----> {60}:
minDownshiftSnrMgn: -----> {60}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {1024000}:
interleaveMaxTxRate: -----> {1536000}:
maxInterleaveDelay: -----> {8}: 4
thresh15MinLofs: -----> {0}:

```

```

thresh15MinLoss: -----> {0}:
thresh15MinLprs: -----> {0}:
thresh15MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: ---> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: -> {0}:
minTxThresholdRateAlarm: --> {0}:
minINP: -----> {0}: 20
phyRSupport: -----> {disable}: enable
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
ginpAdslCpeSupport: -----> {disable}:
ginpAdslCpeEtrMax: -----> {1536}:
ginpAdslCpeEtrMin: -----> {64}:
ginpAdslCpeNdrMax: -----> {1536}:
ginpAdslCpeShineRatio: -----> {10}:
ginpAdslCpeLeftrThreshold: -> {0}:
ginpAdslCpeMaxDelay: -----> {20}:
ginpAdslCpeMinDelay: -----> {0}:
ginpAdslCpeMin: -----> {4}:
ginpAdslCpeMinRSoverhead: --> {0}:
ginpAdslCpeReinCfg: -----> {0}:
ginpAdslCpeReinFreq: -----> {freq120hz}:
ginpAdslCpeRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Configure G.INP parameters

G.INP is a standards based error correction mechanism replacing Phy-R™.



Note: G.INP provides retransmission service on VDSL2 upstream and downstream and on ADSL2+ downstream only.



Note: The G.INP standard does not cover ADSL, and as such, G.INP on ADSL is not supported.

Enabling G.INP

Enable the G.INP support parameter in both the **adsl-co-profile** and the **adsl-cpe-profile**.

- 1 Update the **ginpAdslCoSupport** parameter in the **adsl-co-profile** to enable G.INP.

```

zSH> update adsl-co-profile 1/10/1
adsl-co-profile 1/10/1
Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:

```

```

rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minUpshiftTime: -----> {60}:
minDownshiftTime: -----> {60}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {32736000}:
maxInterleaveDelay: -----> {63}:
interleaveMaxTxRate: -----> {32736000}:
thresh15MinLofs: -----> {0}:
thresh15MinLoss: -----> {0}:
thresh15MinLols: -----> {0}:
thresh15MinLprs: -----> {0}:
thresh15MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: ---> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: -> {0}:
initFailureTrapEnable: ----> {disable}:
reachExtendedAdsl2: -----> {enable}:
minTxThresholdRateAlarm: --> {0}:
minINP: -----> {20}:
phyRSupport: -----> {disable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
txPowerAttenuation: -----> {20}:
cabMode: -----> {0}:
ginpAdslCoSupport: -----> {disable}: enable
ginpAdslCoEtrMax: -----> {32736}:
ginpAdslCoEtrMin: -----> {64}:
ginpAdslCoNdrMax: -----> {32736}:
ginpAdslCoShineRatio: -----> {10}:
ginpAdslCoLeftrThreshold: -> {0}:
ginpAdslCoMaxDelay: -----> {20}:
ginpAdslCoMinDelay: -----> {0}:
ginpAdslCoMin: -----> {4}:
ginpAdslCoMinRSoverhead: --> {0}:
ginpAdslCoReinCfg: -----> {0}:
ginpAdslCoReinFreq: -----> {freq120hz}:
ginpAdslCoRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

2 Update the **ginpAdslCpeSupport** parameter in the adsl-cpe-profile to enable G.INP.

```

zSH> update adsl-cpe-profile 1/10/1
adsl-cpe-profile 1/10/1

```

```

Please provide the following: [q]uit.
rateMode: -----> {adaptatruntime}:
rateChanRatio: -----> {50}:
targetSnrMgn: -----> {60}:
maxSnrMgn: -----> {310}:
minSnrMgn: -----> {0}:
downshiftSnrMgn: -----> {30}:
upshiftSnrMgn: -----> {90}:
minUpshiftSnrMgn: -----> {60}:
minDownshiftSnrMgn: -----> {60}:
fastMinTxRate: -----> {32000}:
interleaveMinTxRate: -----> {32000}:
fastMaxTxRate: -----> {1024000}:
interleaveMaxTxRate: -----> {1536000}:
maxInterleaveDelay: -----> {16}:
thresh15MinLofs: -----> {0}:
thresh15MinLoss: -----> {0}:
thresh15MinLprs: -----> {0}:
thresh15MinESs: -----> {0}:
threshFastRateUp: -----> {0}:
threshInterleaveRateUp: -----> {0}:
threshFastRateDown: -----> {0}:
threshInterleaveRateDown: -----> {0}:
minTxThresholdRateAlarm: -----> {0}:
minINP: -----> {20}:
phyRSupport: -----> {disable}:
phyRmaxINP: -----> {0}:
phyRminRSoverhead: -----> {0}:
phyRRtxRatio: -----> {0}:
ginpAdslCpeSupport: -----> {disable}: enable
ginpAdslCpeEtrMax: -----> {1536}:
ginpAdslCpeEtrMin: -----> {64}:
ginpAdslCpeNdrMax: -----> {1536}:
ginpAdslCpeShineRatio: -----> {10}:
ginpAdslCpeLeftrThreshold: -----> {0}:
ginpAdslCpeMaxDelay: -----> {20}:
ginpAdslCpeMinDelay: -----> {0}:
ginpAdslCpeMin: -----> {4}:
ginpAdslCpeMinRSoverhead: -----> {0}:
ginpAdslCpeReinCfg: -----> {0}:
ginpAdslCpeReinFreq: -----> {freq120hz}:
ginpAdslCpeRtxMode: -----> {preferred}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

ADSL2+ statistics

Verifying the interface

Use the **dslstat** command to display the statistics on a ADSL2+ interface:

```
zSH> dslstat 1-9-1-0/adsl1
```

General Stats:

```

-----
AdminStatus.....UP
LineStatus.....DATA
Line uptime
(DD:HH:MM:SS).....3:19:53:22
DslUpLineRate (bitsPerSec).....1023000
DslDownLineRate (bitsPerSec).....22254800
DslMaxAttainableUpLineRate (bitsPerSec).....1169200
DslMaxAttainableDownLineRate (bitsPerSec)....25560000
Out Octets.....115964
Out Pkts/Cells.....2188
Out Discards.....159
Out Errors.....159
In Octets.....5988788
In Pkts/Cells.....112996
In Discards.....0
In Errors.....0
ATM OCD Count.....0
ATM NCD Count.....0
ATM HEC Count.....3
ATM far-end OCD Count.....0
ATM far-end NCD Count.....0
ATM far-end HEC Count.....0

```

ADSL CPE Information:

```

-----
CPE Serial Number.....
CPE Vendor
Id.....B5004244434D0000A2pB
022g2
CPE Version Number.....A2pB022g2

```

ADSL Physical Stats:

```

-----
Actual Transmission connection standard.....ADSL2+
AdslAtucCurrLineSnrMgn (tenths dB).....95
AdslAtucCurrLineAtn (tenths dB).....0
AdslAtucCurrOutputPwr (tenths dB).....143
AdslAturCurrLineSnrMgn (tenths dB).....66
AdslAturCurrLineAtn (tenths dB).....20
AdslAturCurrOutputPwr (tenths dB).....8
LOFS.....0
LOLS.....0
LOSS.....0
ESS.....6
Inits.....1
Adsl connects.....1
Adsl disconnects.....0

```

near-end statistics:

```

-----
blocks received.....17340235
errored blocks received.....6

```



```

CRC errors on interleaved buffer.....0
CRC errors on fast buffer.....6
FEC corrected errors on interleaved buffer...0
FEC corrected errors on fast buffer.....0
background errored blocks received.....0
non-SES blocks received.....0
Severely Errored Seconds.....0
Unavailable Seconds.....0
Loss of Signal Seconds.....0
Seconds with one/more FECs.....0
Seconds declared as high BER.....0
Fast retrains.....0
Fast retrain failures.....0

```

far-end statistics:

```

-----
blocks received.....17972303
errored blocks received.....33551
CRC errors on interleaved buffer.....0
CRC errors on fast buffer.....11136
FEC corrected errors on interleaved buffer...0
FEC corrected errors on fast buffer.....22415
background errored blocks received.....0
non-SES blocks received.....0
Severely Errored Seconds.....24
Unavailable Seconds.....381834
Loss of Signal Seconds.....24
Seconds with one/more FECs.....323
Loss of Power (dying gasps).....0
Seconds declared as high BER.....22415

```

phyR Statistics:

```

-----
Atuc PhyRActive.....FALSE
Atuc Retransmitted codewords.....0
Atuc Corrected Retransmitted codewords.....0
Atuc UnCorrectableRetransmitted codewords...0
Atur PhyRActive.....FALSE
Atur Retransmitted codewords.....0
Atur Corrected Retransmitted codewords.....0
Atur UnCorrectable Retransmitted codewords...0

```

G.INP Statistics:

```

-----
Atuc ginpActive.....FALSE
Atuc Error Free Throughput Rate (LEFTR) Secs.0
Atuc Error Free Bits.....0
Atuc Minimum Error Free Throughput Rate.....0
Atur ginpActive.....FALSE
Atur Error Free Throughput Rate (LEFTR) Secs.0
Atur Error Free Bits.....0
Atur Minimum Error Free Throughput Rate.....0

```

Clearing DSL counters (ADSL2+)

You can clear DSL counters to make identifying the changing statistics easier to read.

- 1 Clear the statistics using the **dslstat clear** command

```
zSH> dslstat clear 1-9-1-0/adsl
```

- 2 View the changes

After entering the **dslstat** command, (see [Verifying the interface on page 1283](#)) to show the statistics before clearing the statistics. Statistics which are cleared by the **dslstat clear** command are in *bold*.

```
zSH> dslstat 1-9-1-0/adsl
```

General Stats:

```
-----
AdminStatus.....UP
LineStatus.....DATA
DslUpLineRate (bitsPerSec).....1023000
DslDownLineRate (bitsPerSec).....22675400
DslMaxAttainableUpLineRate (bitsPerSec).....1173400
DslMaxAttainableDownLineRate (bitsPerSec)....25656000
In Octets.....742
In Pkts/Cells.....14
In Discards.....0
In Errors.....0
Out
Octets.....1017125120
Out Pkts/Cells.....0
Out Discards.....0
Out Errors.....0
ATM OCD Count.....0
ATM NCD Count.....0
ATM HEC Count.....0
ATM far-end OCD Count.....0
ATM far-end NCD Count.....0
ATM far-end HEC Count.....0
```

ADSL CPE Information:

```
-----
CPE Serial Number.....
CPE Vendor
Id.....B5004244434D0000A2pB
022g2
CPE Version Number.....A2pB022g2
```

ADSL Physical Stats:

```
-----
Actual Transmission connection standard.....ADSL2+
AdslAtucCurrLineSnrMgn (tenths dB).....90
AdslAtucCurrLineAtn (tenths dB).....0
AdslAtucCurrOutputPwr (tenths dB).....143
```

```

AdslAturCurrLineSnrMgn (tenths dB).....59
AdslAturCurrLineAtn (tenths dB).....20
AdslAturCurrOutputPwr (tenths dB).....8
LOFS.....0
LOLS.....0
LOSS.....0
ESS.....0
Inits.....0
Adsl connects.....0
Adsl disconnects.....0

near-end statistics:
-----
blocks received.....330
errored blocks received.....0
CRC errors on interleaved buffer.....0
CRC errors on fast buffer.....0
FEC corrected errors on interleaved buffer...0
FEC corrected errors on fast buffer.....0
background errored blocks received.....0
non-SES blocks received.....0
Severely Errored Seconds.....0
Unavailable Seconds.....0
Loss of Signal Seconds.....0
Seconds with one/more FECs.....0
Seconds declared as high BER.....0
Fast retrains.....0
Fast retrain failures.....0

far-end statistics:
-----
blocks received.....366
errored blocks received.....0
CRC errors on interleaved buffer.....0
CRC errors on fast buffer.....0
FEC corrected errors on interleaved buffer...0
FEC corrected errors on fast buffer.....0
background errored blocks received.....0
non-SES blocks received.....0
Severely Errored Seconds.....0
Unavailable Seconds.....0
Loss of Signal Seconds.....0
Seconds with one/more FECs.....0
Loss of Power (dying gasps).....0
Seconds declared as high BER.....0

```

[Table 159](#) defines the statistics displayed in the **dslstat** command.

Table 156: ADSL2+ statistics

General statistics	Description
AdminStatus	Administrative status of the port: Values: Up Interface is ready to pass packets. Down Interface is unable to pass packets. Testing Interface is in a special testing state and is unable to pass packets.
LineStatus	Provides information about the ADSL link. Values: DOWN — no connection to CPE DOWNLOADING— downloading s/w to CPE DATA — connection established, passing data TEST — loopback or some other test function modem handshake phases (TRAINING, HANDSHAKE, DISCOVERY, ANALYSIS) SELT — selt in progress DELT —delt in progress UNKNOWN — an unknown error occurred (bad s/w, incorrect s/v version. etc.)
Line uptime (DD:HH:MM:SS)	How long the interface has been up in dd hh mm (day, hour, minute, second) format.
DslUpLineRate (bitsPerSec)	Displays the DSL upstream (customer premise > central office) line rate on this interface.
DslDownLineRate (bitsPerSec)	Displays the DSL downstream (central office > customer premise) line rate on this interface.
DslMaxAttainableUpLineRate (bitsPerSec)	Displays the maximum line rate that can be supported on this line in the upstream direction.
DslMaxAttainableDownLineRate (bitsPerSec)	Displays the maximum line rate that can be supported on this line in the downstream direction.
Out Octets	Number of transmitted octets.
Out Pkts/Cells	Number of transmitted packets/cells
Out Discards	Number of transmission discards.
Out Errors	Number of transmission errors.
In Octets	Number of received octets.
In Pkts/Cells	Number of transmitted packets/cells
In Discards	Number of received discards.
In Errors	Number of receive errors.

Table 156: ADSL2+ statistics (Continued)

General statistics	Description
ATM OCD Count	The number Out of Cell Delineation (OCD) events. An Out of Cell Delineation event is defined as seven consecutive ATM cells with Header Error Control (HEC) violations. A high number of these events may indicate a problem with the ATM TC sublayer.
ATM NCD Count	The number of far end No Cell Delineation (NCD) events on the far end.
ATM HEC Count	Number of corrected HEC cells.
ATM far-end OCD Count	The number Out of Cell Delineation (OCD) events. An Out of Cell Delineation event is defined as seven consecutive ATM cells with Header Error Control (HEC) violations. A high number of these events may indicate a problem with the ATM TC sublayer.
ATM far-end NCD Count	The number of far end No Cell Delineation (NCD) events on the far end.
ATM far-end HEC Count	Number of corrected HEC cells at the far-end.
ADSL CPE Information:	
CPE Serial Number	The vendor's serial number for the ADSL CPE device. The displayed information depends on the information the remote modem supplies.
CPE Vendor Id	The vendor portion of the ADSL CPE devices MAC address. The displayed information depends on the information the remote modem supplies.
CPE Version Number	The version number of the software of the ADSL CPE device. The displayed information depends on the information the remote modem supplies.
ADSL Physical Stats:	
Actual Transmission connection standard	Displays the maximum line rate that can be supported on this line in the downstream direction. Values: GHS GDMT T1 GLite Full Rate AutoNegotiate
AdslAtucCurrLineSnrMgn (tenths dB)	SNR Margin is the maximum increase in dB of the noise power received at the ATU-C on upstream direction), such that the BER requirements are met for all bearer channels received at the ATU. It ranges from 640 to 630 units of 0.1 dB (Physical values are -64 to 63 dB).

Table 156: ADSL2+ statistics (Continued)

General statistics	Description
AdslAtucCurrLineAtn (tenths dB)	Measured difference in the total power transmitted by the peer ATU-C and the total power received by this ATU.
AdslAtucCurrOutputPwr (tenths dB)	Actual Aggregate Transmit Power from the ATU-C on upstream direction at the instant of measurement. It ranges from -310 to 310 units of 0.1 dB (Physical values are -31 to 31 dBm).
AdslAturCurrLineSnrMgn (tenths dB)	SNR Margin is the maximum increase in dB of the noise power received at the ATU (ATU-R on downstream direction, such that the BER requirements are met for all bearer channels received at the ATU. It ranges from 640 to 630 units of 0.1 dB (Physical values are -64 to 63 dB).
AdslAturCurrLineAtn (tenths dB)	Measured difference in the total power transmitted by the peer ATU-R and the total power received by this ATU.
AdslAturCurrOutputPwr (tenths dB)	Actual Aggregate Transmit Power from the ATU (ATU-R on downstream direction at the instant of measurement. It ranges from -310 to 310 units of 0.1 dB (Physical values are -31 to 31 dBm).
LOFS	Number of Loss of Frame Seconds.
LOLS	Number of Loss of Line Seconds.
LOSS	Number of Loss of Signal Seconds.
ESS	Number of errored seconds (the number of one-second intervals containing one or more CRC anomalies or one or more LoS or Sef defects) that has been reported in the current 15-minute interval.
Inits	Number of line initialization attempts, including both successful and failed attempts.
Adsl connects	Number of successful connects at the near end since the agent reset.
Adsl disconnects	Number of disconnects at the near end since the agent reset.
near-end statistics:	
blocks received	Number of received blocks at the near end. This statistic is not incremented while there is a UAS or SES error on the interface.
errored blocks received	Number of background errored blocks at the near end. A background block error is an errored block that does not occur as part of a SES. A block refers to Reed Solomon error correction blocks. They are typically 255 bytes of data, and may span several symbols of data, regardless of how many or what parts of ATM cells they represent. This statistic is not incremented while there is a UAS or SES error on the interface.

Table 156: ADSL2+ statistics (Continued)

General statistics	Description
CRC errors on interleaved buffer	Number of cyclic redundancy check (CRC) errors on interleaved buffer at the near end. This statistic is not incremented while there is a UAS or SES error on the interface.
CRC errors on fast buffer	Number of cyclic redundancy check (CRC) errors on fast buffer at the near end. This statistic is not incremented while there is a UAS or SES error on the interface.
FEC corrected errors on interleaved buffer	Number of forward error corrections (FECs) on interleaved buffer at the near end. Forward error correction (Reed Solomon) is applied to the transported data. This process obtains coding gain, resulting in the link requiring lower signal-to-noise ratios (SNRs) for a given data and error rate. This process allows an increase in the data rate under given loop conditions. In addition, interleaving can be applied on top of error correction to obtain a higher degree of protection against error bursts or temporary loss of the data signal. The interleave distributes the data errors over multiple symbols. This action is intended to reduce the number of errors per Reed Solomon codeword to a number within the correction capability of the code. This statistic is not incremented while there is a UAS or SES error on the interface.
FEC corrected errors on fast buffer	Number of forward error corrections (FECs) on fast buffer at the near end. This statistic is not incremented while there is a UAS or SES error on the interface. Fast Buffer—Each ADSL frame consists of two parts, one from each of two buffers: the fast buffer and the interleaved buffer. The fast buffer, in addition to user data, may contain CRC error checking bits, and forward error correcting bits. The fast byte in frame 1, 34, and 35 contain indicator bits used for administrative functions. The interleaved buffer contains purely data.
background errored blocks received	Background errored blocks at near end. A background block error is an errored block that does not occur as part of a SES. A block refers to Reed Solomon error correction blocks. They are typically 255 bytes of data, and may span several symbols of data, regardless of how many or what parts of ATM cells they represent. This statistic is not incremented while there is a UAS or SES error on the interface.
non-SES blocks received	Number of non severely errored seconds (SES) blocks received at the near end.

Table 156: ADSL2+ statistics (Continued)

General statistics	Description
Severely Errored Seconds	Number of severely errored seconds (SES) at the near end. This is the number of 1-second intervals with any of the following error conditions: 18 or more CRC-8 anomalies over all received channels). If a CRC occurs over multiple bearer channels, then each related CRC-8 anomaly is counted only once for the whole set of bearer channels over which the CRC is applied. one or more LOS defects one or more SEF defects one or more LPR defects
Unavailable Seconds	Number of unavailable seconds (UAS) at the near end. This is the number of 1-second intervals for which the ADSL line is unavailable. The ADSL line becomes unavailable after the onset of 10 consecutive severely errored seconds (SESs). Note that the 10 SESs are included in unavailable time. The ADSL line becomes available after 10 consecutive seconds with no SESs. Note that the 10 seconds with no SESs are excluded from unavailable time.
Loss of Signal Seconds	Retrieved via OAM. Count of 1-second intervals containing one or more near end loss of signal (LOS) defects. An LOS failure is declared for either of the following reasons: after 2.5 ± 0.5 seconds of continuous LOS defects if LOS defect is present when a LOF occurs. A line circuit reports a LOS defect when the received power has fallen below the threshold. The threshold is set at 6 dB below the reference power. A LOS failure is cleared after 10 ± 0.5 seconds of no LOS defects. The most common cause for this fault is that the customer premises equipment (CPE) has been turned off. Supported for ADSL2/ADSL2plus only. This statistic is not incremented while there is a UAS or SES error on the interface.
Seconds with one/more FECs	Number of seconds with one or more forward error corrections (FECs) at the near end. These blocks are passed on as good data. This statistic is not incremented while there is a UAS or SES error on the interface.
Seconds declared as high BER	Number of seconds with high Bit Error Rate (BER).
Fast retrains	Number of fast retrains.
Fast retrain failures	Number of fast retrain failures.
far-end statistics:	

Table 156: ADSL2+ statistics (Continued)

General statistics	Description
blocks received	Number of received blocks at the far end. This statistic is not incremented while there is a UAS or SES error on the interface.
errored blocks received	Number of background errored blocks at the far end. A background block error is an errored block that does not occur as part of a SES. A block refers to Reed Solomon error correction blocks. They are typically 255 bytes of data, and may span several symbols of data, regardless of how many or what parts of ATM cells they represent. This statistic is not incremented while there is a UAS or SES error on the interface.
CRC errors on interleaved buffer	Number of cyclic redundancy check (CRC) errors on interleaved buffer at the far end. This statistic is not incremented while there is a UAS or SES error on the interface.
CRC errors on fast buffer	Number of cyclic redundancy check (CRC) errors on fast buffer at the far end. This statistic is not incremented while there is a UAS or SES error on the interface.
FEC corrected errors on interleaved buffer	Number of forward error corrections (FECs) on interleaved buffer at the far end. Forward error correction (Reed Solomon) is applied to the transported data. This process obtains coding gain, resulting in the link requiring lower signal-to-noise ratios (SNRs) for a given data rate and error rate. This process allows an increase in the data rate under given loop conditions. In addition, interleaving can be applied on top of error correction to obtain a higher degree of protection against error bursts or temporary loss of the data signal. The interleaver distributes the data errors over multiple symbols. This action is intended to reduce the number of errors per Reed Solomon codeword to a number within the correction capability of the code. This statistic is not incremented while there is a UAS or SES error on the interface.

Table 156: ADSL2+ statistics (Continued)

General statistics	Description
FEC corrected errors on fast buffer	Number of forward error corrections (FECs) on fast buffer at the far end. This statistic is not incremented while there is a UAS or SES error on the interface. Fast Buffer—Each ADSL frame consists of two parts, one from each of two buffers: the fast buffer and the interleaved buffer. The fast buffer, in addition to user data, may contain CRC error checking bits, and forward error correcting bits. The fast byte in frame 1, 34, and 35 contain indicator bits used for administrative functions. The interleaved buffer contains purely data.
background errored blocks received	Number of background errored blocks at the far end. A background block error is an errored block that does not occur as part of a SES. A block refers to Reed Solomon error correction blocks. They are typically 255 bytes of data, and may span several symbols of data, regardless of how many or what parts of ATM cells they represent. This statistic is not incremented while there is a UAS or SES error on the interface.
non-SES blocks received	Number of non severely errored seconds (SES) blocks received at the far end.
Severely Errored Seconds	Number of errored seconds (the number of one-second intervals containing one or more cyclic redundancy check [CRC] anomalies or one or more loss of signal [LOS] defects) that has been reported in the current 15-minute interval. This statistic is not incremented while there is a UAS or SES error on the interface.
Unavailable Seconds	Number of unavailable seconds (UAS) at the far end. This is the number of 1-second intervals for which the ADSL line is unavailable. The ADSL line becomes unavailable after the onset of 10 consecutive severely errored seconds (SESS). Note that the 10 SESSs are included in unavailable time. The ADSL line becomes available after 10 consecutive seconds with no SESSs. Note that the 10 seconds with no SESSs are excluded from unavailable time.
Loss of Signal Seconds	Loss of signal seconds at the near end.
Seconds with one/more FECs	Number of seconds with one or more forward error corrections (FECs) at the far end. These blocks are passed on as good data. This statistic is not incremented while there is a UAS or SES error on the interface.
Loss of Power (dying gasps)	The ATU-R (remote) device sends a dying-gasp message before it goes down so that the ATU-C (central office) device can differentiate between line down and ATU-R device down events.

Table 156: ADSL2+ statistics (Continued)

General statistics	Description
Seconds declared as high BER	Seconds declared as high BER by the far end.
phyR Statistics:	
Atuc PhyRActive	Is this feature active.
Atuc Retransmitted codewords	ATUC Retransmitted Codewords.
Atuc Corrected Retransmitted codewords	ATUC Retransmitted corrected Codewords.
Atuc UnCorrectableRetransmitted codewords	ATUC Retransmitted uncorrectable Codewords.
Atur Retransmitted codewords	ATUR Retransmitted Codewords.
Atur Corrected Retransmitted codewords	ATUR Retransmitted corrected Codewords.
Atur UnCorrectable Retransmitted codewords	ATUR Retransmitted uncorrectable Codewords.
G.INP Statistics:	
Atuc ginpActive	G.INP/ITU-G.998.4 feature active.
Atuc Error Free Throughput Rate (LEFTR) Secs	This counter counts the number of seconds that experienced a Low Error Free Throughput Rate (LEFTR), i.e., seconds during which the Error Free Throughput dropped below the configured threshold.
Atuc Error Free Bits	This counter counts the number of bits that crossed the alpha1/beta1 interface (bits available to carry user payload).
Atuc Minimum Error Free Throughput Rate	This performance monitoring parameter records the lowest value of Error Free Throughput during the current interval.
Atur ginpActive	G.INP/ITU-G.998.4 feature active.
Atur Error Free Throughput Rate (LEFTR) Secs	This counter counts the number of seconds that experienced a Low Error Free Throughput Rate (LEFTR), i.e., seconds during which the Error Free Throughput dropped below the configured threshold.
Atur Error Free Bits	This counter counts the number of bits that crossed the alpha1/beta1 interface (bits available to carry user payload).
Atur Minimum Error Free Throughput Rate	This performance monitoring parameter records the lowest value of Error Free Throughput during the current interval.

ADSL2+ 48-port bonding

The MXK ADSL2+ 48-port line cards support ADSL2+ bonding using the `bond add group` and the `bond add member` commands.

Bonding allows multiple lines to work together as a single line. Each bonding group can have:

- Two members per bond group.
- Members of a bond group must be contiguous ports which do not cross chip core boundaries. Each chip core has six ports (ports 1-6, 7-12, 13-18, 19-24, and so on). You can add port 1 and 2, 2 and 3, 3 and 4, 4 and 5, 5 and 6, but you cannot combine ports 6 and 7 because that would cross a chip core boundary.
- Bond group numbers must be in appropriate ranges. When using CLI to create a `gbond` group, the valid range for a group is from 1-48. Using ZMS to create a `gbond` group the valid range is from 1-48.

If you attempt to add more than two members, non-contiguous ports, ports which cross chip boundaries, or groups outside of the valid range the CLI will remind you of these rules. You also cannot add the same member to different bond groups.

Creating a `gbond` group on an ADSL2+ card

Create a single *gbond* group by first creating the bond group, then adding the members of the *gbond* group.

- 1 Create the *gbond* group with the `bond add group` command:

```
zSH> bond add group 1-9-1-0/gbond
```

- 2 Add members to the *gbond* group with the `bond add member` command:

```
zSH> bond add member 1-9-1-0/gbond 1-9-1-0/adsl
```

```
zSH> bond add member 1-9-1-0/gbond 1-9-2-0/adsl
```

- 3 View the bond group and the bond group members with the **`bond show group`** command:

```
zSH> bond show group 1-9-1-0/gbond
```

Bond Groups				
Slot	GrpId	Type	State	Name
9	1	gbond	OOS	1-9-1-0
Group Members				
Slot	Port	Type	State	Name
9	1	adsl	OOS	1-9-1-0
9	2	adsl	OOS	1-9-2-0

View which *gbond* groups exist on a particular slot with the **`bond group slot`** command:

```
zSH> bond show slot 9
```

Bond Groups				
Slot	GrpId	Type	State	Name
9	1	gbond	OOS	1-9-1-0

The *gbond* group is displayed but does not show the bond group members.

Deleting a member of a gbond group

Use the bond delete member command to delete a member of a *gbond* group:

```
zSH> bond delete member 1-9-1-0/gbond 1-9-1-0/adsl
```

Deleting a gbond group

Use the bond delete group command to delete a *gbond* group:



Note: All *members* of the bond group must be deleted before the bond group can be deleted.

```
zSH> bond delete group 1-9-1-0/gbond
```

Moving members of a gbond group

After creating two *gbond* groups, a member can be moved from one bond group to another.

- 1 Create a *gbond* group:

```
zSH> bond add group 1-9-10-0/gbond
```

- 2 Add members to the *gbond* group:

```
zSH> bond add member 1-9-10-0/gbond 1-9-1-0/adsl
```

```
zSH> bond add member 1-9-10-0/gbond 1-9-2-0/adsl
```

- 3 View the *gbond* group and the members of the *gbond* group:

```
zSH> bond show group 1-9-10-0/gbond
```

Bond Groups				
Slot	GrpId	Type	State	Name
9	10	gbond	ACT	1-9-10-0
Group Members				
Slot	Port	Type	State	Name
9	1	adsl	ACT	1-9-1-0
9	2	adsl	ACT	1-9-2-0

- 4 Create the next *gbond* group:

```
zSH> bond add group 1-9-11-0/gbond
```

5 Add a member to the *gbond* group:

```
zSH> bond add member 1-9-11-0/gbond 1-9-3-0/adsl
```

6 View existing *gbond* groups:

```
zSH> bond show all
```

```

Bond Groups
Slot  GrpId  Type      State      Name
  9      11   gbond     ACT        1-9-11-0
  9      10   gbond     ACT        1-9-10-0

```

View the *gbond* groups and their members:

```
zSH> bond show group 1-9-10-0/gbond
```

```

Bond Groups
Slot  GrpId  Type      State      Name
  9      10   gbond     ACT        1-9-10-0
Group Members
Slot  Port   Type      State      Name
  9      1    adsl     ACT        1-9-1-0
  9      2    adsl     ACT        1-9-2-0

```

```
zSH> bond show group 1-9-11-0/gbond
```

```

Bond Groups
Slot  GrpId  Type      State      Name
  9      11   gbond     ACT        1-9-11-0
Group Members
Slot  Port   Type      State      Name
  9      3    adsl     ACT        1-9-3-0

```

7 Move a member from *gbond* group 1-9-10-0/*gbond* to 1-9-11-0/*gbond*:

```
zSH> bond move 1-9-10-0/gbond 1-9-11-0/gbond 1-9-2-0/adsl
```

8 View the members in *gbond* group 1-9-10-0/*gbond* and 1-9-11-0/*gbond*:

```
zSH> bond show group 1-9-10-0/gbond
```

```

Bond Groups
Slot  GrpId  Type      State      Name
  9      10   gbond     ACT        1-9-10-0
Group Members
Slot  Port   Type      State      Name
  9      1    adsl     ACT        1-9-1-0

```

```
zSH> bond show group 1-9-11-0/gbond
```

```

Bond Groups
Slot  GrpId  Type      State      Name
  9      11   gbond     ACT        1-9-11-0
Group Members

```

Slot	Port	Type	State	Name
9	3	adsl	ACT	1-9-3-0
9	2	adsl	ACT	1-9-2-0

Member *1-9-2-0/adsl* moved from group *1-9-10-0/gbond* to *1-9-11-0/gbond*.

ADSL2+ 72-port bonding

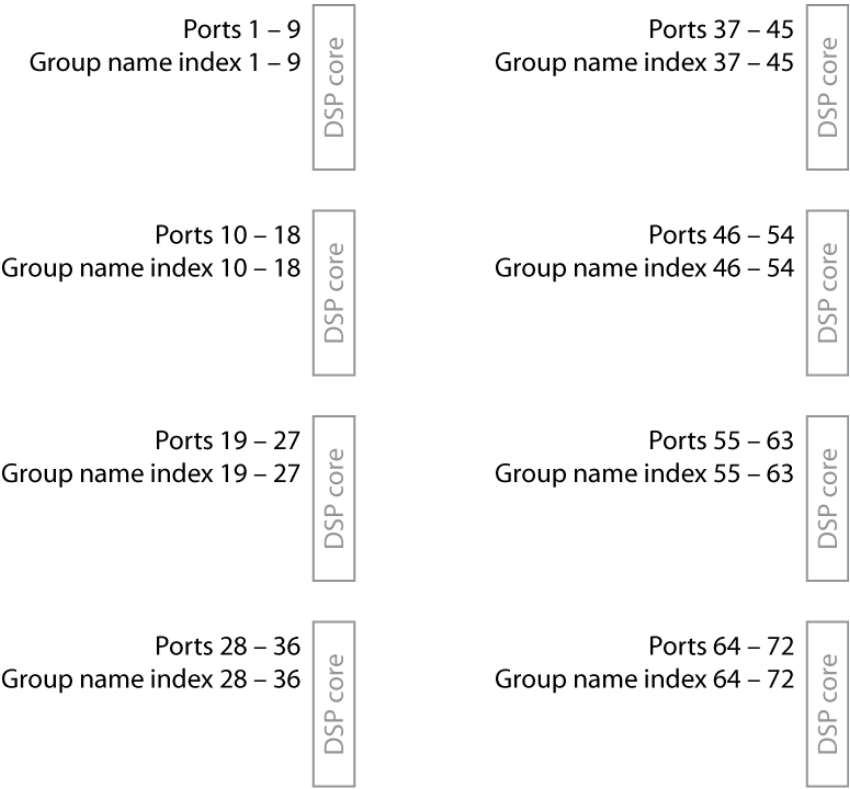
The MXK ADSL2+ 72-port line card supports ADSL2+ bonding using the bond add group and the bond add member commands.

Bonding allows multiple lines to work together as a single line. Each bonding group can have:

- Two members per bond group.
- Members of a *gbond* group *must* be in a bond group that does not cross chip core boundary (see [Figure 158](#)). There are 8 DSP cores, so 64 ports can be bonded.

Each chip core has nine ports (ports 1-9, 10-18, 19-27, 28-36, and so on). You can create *gbond* groups with any combination of eight of the nine ports in a chip core, but you cannot combine ports 9 and 10 because that would cross a chip core boundary.

Figure 158: 72-port ADSL DSP core boundaries



- The *gbond* group index *must* match the ports of the chip core.
- Bond group numbers must be in appropriate ranges. When using CLI to create a *gbond* group, the valid range for a group is from 1-72. Using ZMS to create a *gbond* group the valid range is from 1-72.

When configuring *gbond* groups, if you attempt to add more than two members, ports which cross chip boundaries, or groups outside of the valid

range, the CLI will remind you of these rules. Also, you cannot add a member to more than one *gbond* group.

Create *gbond* groups on 72-port ADSL cards

Creating *gbond* groups on an ADSL2+ 72-port card

Create a single *gbond* group by first creating the bond group, then adding the members of the *gbond* group.

- 1 Create the *gbond* group with the **bond add group** command.

```
zSH> bond add group 1-2-1-0/gbond
```

- 2 Add members to the *gbond* group with the **bond add member** command.

```
zSH> bond add member 1-2-1-0/gbond 1-2-1-0/adsl
```

```
zSH> bond add member 1-2-1-0/gbond 1-2-2-0/adsl
```

- 3 View the *gbond* group and *gbond* group members.

```
zSH> bond show group 1-2-1-0/gbond
```

```

Bond Groups
Slot  GrpId  Type      State   Name      Desc
  2      1   gbond     OOS     1-2-1-0   -

Group Members
Slot  Port   Type      State   Name      Desc
  2    1   adsl      OOS     1-2-1-0   -
  2    2   adsl      OOS     1-2-2-0   -

```

- 4 Create the next *gbond* group.

```
zSH> bond add group 1-2-2-0/gbond
```

- 5 Add members to the *gbond* group.

```
zSH> bond add member 1-2-2-0/gbond 1-2-3-0/adsl
```

```
zSH> bond add member 1-2-2-0/gbond 1-2-4-0/adsl
```

- 6 View the *gbond* group.

```
zSH> bond show group 1-2-2-0/gbond
```

```

Bond Groups
Slot  GrpId  Type      State   Name      Desc
  2      2   gbond     ACT     1-2-2-0   -

Group Members
Slot  Port   Type      State   Name      Desc
  2    3   adsl      ACT     1-2-3-0   -
  2    4   adsl      ACT     1-2-4-0   -

```

Viewing all existing bond groups

View all existing bond groups.

```
zSH> bond show all
```

		Bond Groups			
Slot	GrpId	Type	State	Name	Desc
2	2	gbond	ACT	1-2-2-0	-
2	1	gbond	ACT	1-2-1-0	-

Delete bond groups

Deleting a bond group

Use the **bond delete member** command to delete a member of a *gbond* group.



Note: All *members* of the bond group must be deleted before the bond group can be deleted.

- 1 Delete the *gbond* group members.

```
zSH> bond delete member 1-2-2-0/gbond 1-2-3-0/adsl
```

```
zSH> bond delete member 1-2-2-0/gbond 1-2-4-0/adsl
```

Caution: group is now empty!

- 2 Delete the *gbond* group.

```
zSH> bond delete group 1-2-2-0/gbond
```

ADSL2+ POTS line card ATM

ATM data

The MXK MXK-ADSL2+-POTS-BCM-48A-2S line card communicates with subscriber integrated access devices (IADs) or DSL modems using ATM over DSL interfaces. The card relays the traffic to the uplink port, which provides a high-speed interface to an Ethernet network.

VPI and VCI ranges

[Table 157](#) lists the VPI/VCI support for the MXK. Note that VPI/VCI ranges can be changed.

Table 157: MXK VPI/VCI ranges

Interface	Default ranges
ADSL (per port)	VPI: 0 to 15 VCI: 32 to 255

Service categories

The MXK supports the following ATM service categories:

- Constant Bit Rate (CBR)
- non-real-time variable bit rate (nrt-VBR)
- real-time variable bit rate (rt-VBR)
- unspecified bit rate (UBR)

Constant Bit Rate (CBR)

The CBR service class is designed for ATM virtual circuits (VCs) needing a static amount of bandwidth that is continuously available for the duration of the active connection. An ATM VC configured as CBR can send cells at peak cell rate (PCR) at any time and for any duration. It can also send cells at a rate less than the PCR or even emit no cells.

Non-real-time variable bit rate (nrt-VBR)

The nrt-VBR service category is used by applications that are tolerant of network delays and do not require a timing relationship on each side of the connection. The nrt-VBR service supports somewhat bursty connections having less-stringent delay requirements than rt-VBR, but still require low cell loss. The source traffic descriptor is characterized by peak cell rate (PCR). nrt-VBR and UBR have the same priority in the MXK.

Real-time variable bit rate (rt-VBR)

The rt-VBR service category is used by applications that require a tightly constrained delay and delay variation. The source traffic descriptor is characterized by peak cell rate (PCR). rt-VBR has the highest priority in the MXK.

Unspecified bit rate (UBR)

The UBR service category does not specify traffic-related guarantees. No numerical commitments are made with respect to the cell loss ratio (CLR) experienced by the connection, or the cell transfer delay (CTD) experienced by the cells. With UBR service, the available bandwidth is fairly distributed to the active UBR subscribers. nrt-VBR and UBR have the same priority in the MXK.

Traffic descriptors

Each ATM endpoint requires a traffic descriptor, which defines the traffic parameters and type of service provided on ATM interfaces. Traffic descriptors are configured in **atm-traf-descr** records.



Note: ATM traffic policing and shaping are only supported in the downstream direction.

Quality of Service (QoS) parameters such as max cell transfer delay (maxCTD) and cell loss ratio (CLR) do not apply to a single node on the network and so are not provisioned for individual VCs.

Traffic descriptor parameters

Table 158 shows the required parameters used to define MXK traffic descriptors.

Table 158: ATM traffic descriptor parameters

TD type	td_param1	td_param2	td_param3	td_param4	td_param5
atmNoClpNoScr	PCR for CLP=0+1 traffic must be > 0	Not used	Not used	Not used	Not used



Tip: Refer to the following specifications for more information about traffic descriptors:

- ATM Forum, ATM User-Network Interface, Version 3.0 (UNI 3.0) Specification, 1994.
- ATM Forum, ATM User-Network Interface, Version 3.1 (UNI 3.1) Specification, November 1994.

ATM sample configurations

This section provides two ATM sample configurations, one for data and one for video applications.

ATM traffic descriptor example for data

Example for 1.5 Mbps downstream rate for data application.

```
zSH> new atm-traf-descr 1
atm-traf-descr 1
Please provide the following: [q]uit.
td_type: -----> {atmNoClpNoScr}:
td_param1: -----> {0}: 3661
td_param2: -----> {0}:
td_param3: -----> {0}:
td_param4: -----> {0}:
td_param5: -----> {0}:
cac-divider: -----> {1}:
td_service_category: -----> {ubr}: ubr
usage-parameter-control: -> {true}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

ATM traffic descriptor example for video

Example for approximately 16 Mbps stream per port for video application.

```
zSH> new atm-traf-descr 1
atm-traf-descr 1
Please provide the following: [q]uit.
td_type: -----> {atmNoClpNoScr}:
td_param1: -----> {0}: 34736
td_param2: -----> {0}:
td_param3: -----> {0}:
td_param4: -----> {0}:
td_param5: -----> {0}:
cac-divider: -----> {1}:
td_service_category: -----> {ubr}: ubr
usage-parameter-control: -> {true}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

ATM statistics

Real-time ATM statistics on the MXK are provided through the NetHorizon ZMS client.

The ZMS performance manager periodically collects real-time statistical data. You can monitor real-time data at a polling interval of your choice. For

information on how to access ZMS ATM statistics, refer to *the NetHorizon User's Guide* and the NetHorizon online help.

ADSL2+ statistics

Verifying the interface

Use the **dslstat** command to display the statistics on a ADSL2+ interface:

```
zSH> dslstat 1-9-1-0/adsl
General Stats:
-----
AdminStatus.....UP
LineStatus.....DATA
Line uptime
(DD:HH:MM:SS).....3:19:53:22
DslUpLineRate (bitsPerSec).....1023000
DslDownLineRate (bitsPerSec).....22254800
DslMaxAttainableUpLineRate (bitsPerSec).....1169200
DslMaxAttainableDownLineRate (bitsPerSec)....25560000
Out Octets.....115964
Out Pkts/Cells.....2188
Out Discards.....159
Out Errors.....159
In Octets.....5988788
In Pkts/Cells.....112996
In Discards.....0
In Errors.....0
ATM OCD Count.....0
ATM NCD Count.....0
ATM HEC Count.....3
ATM far-end OCD Count.....0
ATM far-end NCD Count.....0
ATM far-end HEC Count.....0

ADSL CPE Information:
-----
CPE Serial Number.....
CPE Vendor
Id.....B5004244434D0000A2pB
022g2
CPE Version Number.....A2pB022g2

ADSL Physical Stats:
-----
Actual Transmission connection standard.....ADSL2+
AdslAtucCurrLineSnrMgn (tenths dB).....95
AdslAtucCurrLineAtn (tenths dB).....0
AdslAtucCurrOutputPwr (tenths dB).....143
AdslAturCurrLineSnrMgn (tenths dB).....66
AdslAturCurrLineAtn (tenths dB).....20
AdslAturCurrOutputPwr (tenths dB).....8
LOFS.....0
LOLS.....0
LOSS.....0
ESS.....6
```

```

Inits.....1
Adsl connects.....1
Adsl disconnects.....0

near-end statistics:
-----
blocks received.....17340235
errored blocks received.....6
CRC errors on interleaved buffer.....0
CRC errors on fast buffer.....6
FEC corrected errors on interleaved buffer...0
FEC corrected errors on fast buffer.....0
background errored blocks received.....0
non-SES blocks received.....0
Severely Errored Seconds.....0
Unavailable Seconds.....0
Loss of Signal Seconds.....0
Seconds with one/more FECs.....0
Seconds declared as high BER.....0
Fast retrains.....0
Fast retrain failures.....0

far-end statistics:
-----
blocks received.....17972303
errored blocks received.....33551
CRC errors on interleaved buffer.....0
CRC errors on fast buffer.....11136
FEC corrected errors on interleaved buffer...0
FEC corrected errors on fast buffer.....22415
background errored blocks received.....0
non-SES blocks received.....0
Severely Errored Seconds.....24
Unavailable Seconds.....381834
Loss of Signal Seconds.....24
Seconds with one/more FECs.....323
Loss of Power (dying gasps).....0
Seconds declared as high BER.....22415

phyR Statistics:
-----
Atuc Retransmitted codewords.....0
Atuc Corrected Retransmitted codewords.....0
Atuc UnCorrectableRetransmitted codewords...0
Atur Retransmitted codewords.....0
Atur Corrected Retransmitted codewords.....0
Atur UnCorrectable Retransmitted codewords...0

```

Clearing DSL counters (ADSL2+)

You can clear DSL counters to make identifying the changing statistics easier to read.

- 1 Clear the statistics using the **dslstat clear** command


```
zSH> dslstat clear 1-9-1-0/adsl
```

2 View the changes

After entering the **dslstat** command, (see [Verifying the interface on page 1283](#)) to show the statistics before clearing the statistics. Statistic which are cleared by the **dslstat clear** command are in *bold*.

```
zSH> dslstat 1-9-1-0/adsl
```

General Stats:

```
-----
AdminStatus.....UP
LineStatus.....DATA
Line uptime
(DD:HH:MM:SS).....0:01:07:48
DslUpLineRate (bitsPerSec).....1023000
DslDownLineRate (bitsPerSec).....22675400
DslMaxAttainableUpLineRate (bitsPerSec).....1173400
DslMaxAttainableDownLineRate (bitsPerSec)....25656000
Out
Octets.....1017125120
Out Pkts/Cells.....0
Out Discards.....0
Out Errors.....0
In Octets.....742
In Pkts/Cells.....14
In Discards.....0
In Errors.....0
ATM OCD Count.....0
ATM NCD Count.....0
ATM HEC Count.....0
ATM far-end OCD Count.....0
ATM far-end NCD Count.....0
ATM far-end HEC Count.....0
```

ADSL CPE Information:

```
-----
CPE Serial Number.....
CPE Vendor
Id.....B5004244434D0000A2pB
022g2
CPE Version Number.....A2pB022g2
```

ADSL Physical Stats:

```
-----
Actual Transmission connection standard.....ADSL2+
AdslAtucCurrLineSnrMgn (tenths dB).....90
AdslAtucCurrLineAtn (tenths dB).....0
AdslAtucCurrOutputPwr (tenths dB).....143
AdslAturCurrLineSnrMgn (tenths dB).....59
AdslAturCurrLineAtn (tenths dB).....20
AdslAturCurrOutputPwr (tenths dB).....8
LOFS.....0
```

LOLS.....	0
LOSS.....	0
ESS.....	0
Inits.....	0
Adsl connects.....	0
Adsl disconnects.....	0
near-end statistics:	

blocks received.....	330
errored blocks received.....	0
CRC errors on interleaved buffer.....	0
CRC errors on fast buffer.....	0
FEC corrected errors on interleaved buffer...	0
FEC corrected errors on fast buffer.....	0
background errored blocks received.....	0
non-SES blocks received.....	0
Severely Errored Seconds.....	0
Unavailable Seconds.....	0
Loss of Signal Seconds.....	0
Seconds with one/more FECs.....	0
Seconds declared as high BER.....	0
Fast retrains.....	0
Fast retrain failures.....	0
far-end statistics:	

blocks received.....	366
errored blocks received.....	0
CRC errors on interleaved buffer.....	0
CRC errors on fast buffer.....	0
FEC corrected errors on interleaved buffer...	0
FEC corrected errors on fast buffer.....	0
background errored blocks received.....	0
non-SES blocks received.....	0
Severely Errored Seconds.....	0
Unavailable Seconds.....	0
Loss of Signal Seconds.....	0
Seconds with one/more FECs.....	0
Loss of Power (dying gasps).....	0
Seconds declared as high BER.....	0

Table 159 defines the statistics displayed in the **dslstat** command.

Table 159: ADSL2+ statistics

General statistics	Description
AdminStatus	Administrative status of the port: Values: Up Interface is ready to pass packets. Down Interface is unable to pass packets. Testing Interface is in a special testing state and is unable to pass packets.
LineStatus	Provides information about the ADSL link. Values: DOWN — no connection to CPE DOWNLOADING— downloading s/w to CPE DATA — connection established, passing data TEST — loopback or some other test function modem handshake phases (TRAINING, HANDSHAKE, DISCOVERY, ANALYSIS) SELT — selt in progress DELT —delt in progress UNKNOWN — an unknown error occurred (bad s/w, incorrect s/v version. etc.)
Line uptime (DD:HH:MM:SS)	How long the interface has been up in dd hh mm (day, hour, minute, second) format.
DslUpLineRate (bitsPerSec)	Displays the DSL upstream (customer premise > central office) line rate on this interface.
DslDownLineRate (bitsPerSec)	Displays the DSL downstream (central office > customer premise) line rate on this interface.
DslMaxAttainableUpLineRate (bitsPerSec)	Displays the maximum line rate that can be supported on this line in the upstream direction.
DslMaxAttainableDownLineRate (bitsPerSec)	Displays the maximum line rate that can be supported on this line in the downstream direction.
Out Octets	Number of transmitted octets.
Out Pkts/Cells	Number of transmitted packets/cells
Out Discards	Number of transmission discards.
Out Errors	Number of transmission errors.
In Octets	Number of received octets.
In Pkts/Cells	Number of transmitted packets/cells
In Discards	Number of received discards.
In Errors	Number of receive errors.

Table 159: ADSL2+ statistics (Continued)

General statistics	Description
ATM OCD Count	The number Out of Cell Delineation (OCD) events. An Out of Cell Delineation event is defined as seven consecutive ATM cells with Header Error Control (HEC) violations. A high number of these events may indicate a problem with the ATM TC sublayer.
ATM NCD Count	The number of far end No Cell Delineation (NCD) events on the far end.
ATM HEC Count	Number of corrected HEC cells.
ATM far-end OCD Count	The number Out of Cell Delineation (OCD) events. An Out of Cell Delineation event is defined as seven consecutive ATM cells with Header Error Control (HEC) violations. A high number of these events may indicate a problem with the ATM TC sublayer.
ATM far-end NCD Count	The number of far end No Cell Delineation (NCD) events on the far end.
ATM far-end HEC Count	Number of corrected HEC cells at the far-end.
ADSL CPE Information:	
CPE Serial Number	The vendor's serial number for the ADSL CPE device. The displayed information depends on the information the remote modem supplies.
CPE Vendor Id	The vendor portion of the ADSL CPE devices MAC address. The displayed information depends on the information the remote modem supplies.
CPE Version Number	The version number of the software of the ADSL CPE device. The displayed information depends on the information the remote modem supplies.
ADSL Physical Stats:	
Actual Transmission connection standard	Displays the maximum line rate that can be supported on this line in the downstream direction. Values: GHS GDMT T1 GLite Full Rate AutoNegotiate
AdslAtucCurrLineSnrMgn (tenths dB)	SNR Margin is the maximum increase in dB of the noise power received at the ATU-C on upstream direction), such that the BER requirements are met for all bearer channels received at the ATU. It ranges from 640 to 630 units of 0.1 dB (Physical values are -64 to 63 dB).

Table 159: ADSL2+ statistics (Continued)

General statistics	Description
AdslAtucCurrLineAtn (tenths dB)	Measured difference in the total power transmitted by the peer ATU-C and the total power received by this ATU.
AdslAtucCurrOutputPwr (tenths dB)	Actual Aggregate Transmit Power from the ATU-C on upstream direction at the instant of measurement. It ranges from -310 to 310 units of 0.1 dB (Physical values are -31 to 31 dBm).
AdslAturCurrLineSnrMgn (tenths dB)	SNR Margin is the maximum increase in dB of the noise power received at the ATU (ATU-R on downstream direction, such that the BER requirements are met for all bearer channels received at the ATU. It ranges from 640 to 630 units of 0.1 dB (Physical values are -64 to 63 dB).
AdslAturCurrLineAtn (tenths dB)	Measured difference in the total power transmitted by the peer ATU-R and the total power received by this ATU.
AdslAturCurrOutputPwr (tenths dB)	Actual Aggregate Transmit Power from the ATU (ATU-R on downstream direction at the instant of measurement. It ranges from -310 to 310 units of 0.1 dB (Physical values are -31 to 31 dBm).
LOFS	Number of Loss of Frame Seconds.
LOLS	Number of Loss of Line Seconds.
LOSS	Number of Loss of Signal Seconds.
ESS	Number of errored seconds (the number of one-second intervals containing one or more CRC anomalies or one or more LoS or Sef defects) that has been reported in the current 15-minute interval.
Inits	Number of line initialization attempts, including both successful and failed attempts.
Adsl connects	Number of successful connects at the near end since the agent reset.
Adsl disconnects	Number of disconnects at the near end since the agent reset.
near-end statistics:	
blocks received	Number of received blocks at the near end. This statistic is not incremented while there is a UAS or SES error on the interface.
errored blocks received	Number of background errored blocks at the near end. A background block error is an errored block that does not occur as part of a SES. A block refers to Reed Solomon error correction blocks. They are typically 255 bytes of data, and may span several symbols of data, regardless of how many or what parts of ATM cells they represent. This statistic is not incremented while there is a UAS or SES error on the interface.

Table 159: ADSL2+ statistics (Continued)

General statistics	Description
CRC errors on interleaved buffer	Number of cyclic redundancy check (CRC) errors on interleaved buffer at the near end. This statistic is not incremented while there is a UAS or SES error on the interface.
CRC errors on fast buffer	Number of cyclic redundancy check (CRC) errors on fast buffer at the near end. This statistic is not incremented while there is a UAS or SES error on the interface.
FEC corrected errors on interleaved buffer	Number of forward error corrections (FECs) on interleaved buffer at the near end. Forward error correction (Reed Solomon) is applied to the transported data. This process obtains coding gain, resulting in the link requiring lower signal-to-noise ratios (SNRs) for a given data and error rate. This process allows an increase in the data rate under given loop conditions. In addition, interleaving can be applied on top of error correction to obtain a higher degree of protection against error bursts or temporary loss of the data signal. The interleave distributes the data errors over multiple symbols. This action is intended to reduce the number of errors per Reed Solomon codeword to a number within the correction capability of the code. This statistic is not incremented while there is a UAS or SES error on the interface.
FEC corrected errors on fast buffer	Number of forward error corrections (FECs) on fast buffer at the near end. This statistic is not incremented while there is a UAS or SES error on the interface. Fast Buffer—Each ADSL frame consists of two parts, one from each of two buffers: the fast buffer and the interleaved buffer. The fast buffer, in addition to user data, may contain CRC error checking bits, and forward error correcting bits. The fast byte in frame 1, 34, and 35 contain indicator bits used for administrative functions. The interleaved buffer contains purely data.
background errored blocks received	Background errored blocks at near end. A background block error is an errored block that does not occur as part of a SES. A block refers to Reed Solomon error correction blocks. They are typically 255 bytes of data, and may span several symbols of data, regardless of how many or what parts of ATM cells they represent. This statistic is not incremented while there is a UAS or SES error on the interface.
non-SES blocks received	Number of non severely errored seconds (SES) blocks received at the near end.

Table 159: ADSL2+ statistics (Continued)

General statistics	Description
Severely Errored Seconds	Number of severely errored seconds (SES) at the near end. This is the number of 1-second intervals with any of the following error conditions: 18 or more CRC-8 anomalies over all received channels). If a CRC occurs over multiple bearer channels, then each related CRC-8 anomaly is counted only once for the whole set of bearer channels over which the CRC is applied. one or more LOS defects one or more SEF defects one or more LPR defects
Unavailable Seconds	Number of unavailable seconds (UAS) at the near end. This is the number of 1-second intervals for which the ADSL line is unavailable. The ADSL line becomes unavailable after the onset of 10 consecutive severely errored seconds (SESs). Note that the 10 SESs are included in unavailable time. The ADSL line becomes available after 10 consecutive seconds with no SESs. Note that the 10 seconds with no SESs are excluded from unavailable time.
Loss of Signal Seconds	Retrieved via OAM. Count of 1-second intervals containing one or more near end loss of signal (LOS) defects. An LOS failure is declared for either of the following reasons: after 2.5 ± 0.5 seconds of continuous LOS defects if LOS defect is present when a LOF occurs. A line circuit reports a LOS defect when the received power has fallen below the threshold. The threshold is set at 6 dB below the reference power. A LOS failure is cleared after 10 ± 0.5 seconds of no LOS defects. The most common cause for this fault is that the customer premises equipment (CPE) has been turned off. Supported for ADSL2/ADSL2plus only. This statistic is not incremented while there is a UAS or SES error on the interface.
Seconds with one/more FECs	Number of seconds with one or more forward error corrections (FECs) at the near end. These blocks are passed on as good data. This statistic is not incremented while there is a UAS or SES error on the interface.
Seconds declared as high BER	Number of seconds with high Bit Error Rate (BER).
Fast retrains	Number of fast retrains.
Fast retrain failures	Number of fast retrain failures.
far-end statistics:	

Table 159: ADSL2+ statistics (Continued)

General statistics	Description
blocks received	Number of received blocks at the far end. This statistic is not incremented while there is a UAS or SES error on the interface.
errored blocks received	Number of background errored blocks at the far end. A background block error is an errored block that does not occur as part of a SES. A block refers to Reed Solomon error correction blocks. They are typically 255 bytes of data, and may span several symbols of data, regardless of how many or what parts of ATM cells they represent. This statistic is not incremented while there is a UAS or SES error on the interface.
CRC errors on interleaved buffer	Number of cyclic redundancy check (CRC) errors on interleaved buffer at the far end. This statistic is not incremented while there is a UAS or SES error on the interface.
CRC errors on fast buffer	Number of cyclic redundancy check (CRC) errors on fast buffer at the far end. This statistic is not incremented while there is a UAS or SES error on the interface.
FEC corrected errors on interleaved buffer	Number of forward error corrections (FECs) on interleaved buffer at the far end. Forward error correction (Reed Solomon) is applied to the transported data. This process obtains coding gain, resulting in the link requiring lower signal-to-noise ratios (SNRs) for a given data rate and error rate. This process allows an increase in the data rate under given loop conditions. In addition, interleaving can be applied on top of error correction to obtain a higher degree of protection against error bursts or temporary loss of the data signal. The interleaver distributes the data errors over multiple symbols. This action is intended to reduce the number of errors per Reed Solomon codeword to a number within the correction capability of the code. This statistic is not incremented while there is a UAS or SES error on the interface.

Table 159: ADSL2+ statistics (Continued)

General statistics	Description
FEC corrected errors on fast buffer	Number of forward error corrections (FECs) on fast buffer at the far end. This statistic is not incremented while there is a UAS or SES error on the interface. Fast Buffer—Each ADSL frame consists of two parts, one from each of two buffers: the fast buffer and the interleaved buffer. The fast buffer, in addition to user data, may contain CRC error checking bits, and forward error correcting bits. The fast byte in frame 1, 34, and 35 contain indicator bits used for administrative functions. The interleaved buffer contains purely data.
background errored blocks received	Number of background errored blocks at the far end. A background block error is an errored block that does not occur as part of a SES. A block refers to Reed Solomon error correction blocks. They are typically 255 bytes of data, and may span several symbols of data, regardless of how many or what parts of ATM cells they represent. This statistic is not incremented while there is a UAS or SES error on the interface.
non-SES blocks received	Number of non severely errored seconds (SES) blocks received at the far end.
Severely Errored Seconds	Number of errored seconds (the number of one-second intervals containing one or more cyclic redundancy check [CRC] anomalies or one or more loss of signal [LOS] defects) that has been reported in the current 15-minute interval. This statistic is not incremented while there is a UAS or SES error on the interface.
Unavailable Seconds	Number of unavailable seconds (UAS) at the far end. This is the number of 1-second intervals for which the ADSL line is unavailable. The ADSL line becomes unavailable after the onset of 10 consecutive severely errored seconds (SESS). Note that the 10 SESSs are included in unavailable time. The ADSL line becomes available after 10 consecutive seconds with no SESSs. Note that the 10 seconds with no SESSs are excluded from unavailable time.
Loss of Signal Seconds	Loss of signal seconds at the near end.
Seconds with one/more FECs	Number of seconds with one or more forward error corrections (FECs) at the far end. These blocks are passed on as good data. This statistic is not incremented while there is a UAS or SES error on the interface.
Loss of Power (dying gasps)	The ATU-R (remote) device sends a dying-gasp message before it goes down so that the ATU-C (central office) device can differentiate between line down and ATU-R device down events.

Table 159: ADSL2+ statistics (Continued)

General statistics	Description
Seconds declared as high BER	Seconds declared as high BER by the far end.
phyR Statistics:	
Atuc Retransmitted codewords	ATUC Retransmitted Codewords.
Atuc Corrected Retransmitted codewords	ATUC Retransmitted corrected Codewords.
Atuc UnCorrectableRetransmitted codewords	ATUC Retransmitted uncorrectable Codewords.
Atur Retransmitted codewords	ATUR Retransmitted Codewords.
Atur Corrected Retransmitted codewords	ATUR Retransmitted corrected Codewords.
Atur UnCorrectable Retransmitted codewords	ATUR Retransmitted uncorrectable Codewords.

ADSL2+ Cabinet Mode

Cabinet mode is supported on the MXK ADSL cards and is normally employed on systems located in outside plant, or street cabinets which share the same binder group as ADSL services originating from the CO. When enabled, the Cabinet Mode feature disables the use of all downstream frequencies below a specified frequency.

While Cabinet mode is beneficial in reducing the disturbances on adjacent ADSL services from the CO, it does so by reducing the frequencies available for ADSL services from the MXK. This reduction of frequencies will diminish the overall rate and reach performance of DSL services from the MXK. Therefore, this feature should only be used in situations where the MXK is adversely affecting the performance of ADSL services from the CO, or when mandated by the incumbent carrier in loop unbundling applications.



Note: For cabinet mode operation, the CPE device on the connection must also support the feature and be configured to respond to a shift in frequencies used during link and line training sequences. If the CPE is not able to respond to shifts in frequencies during link and line training sequences, it may take a long time (if ever) to establish sync between the CPE and the MXK.

Setting cabinet mode

On MXK ADSL cards, cabinet mode may be given a filter setting which defines the cutoff frequency.

For MXK ADSL cards cabinet mode is enabled via the **cabMode** parameter in the **adsl-co-profile**, by setting the value from '1' to '15' which sets the cutoff frequency as show in [Table 160](#),

[MXK ADSL Annex A/M Transmit Filter Settings](#) and [Table 161](#), [MXK ADSL Annex B Transmit Filter Settings](#). Cabinet mode is disabled by setting the parameter to '0.'

Please note from [Table 160](#) and [Table 161](#) that, due to the reduction in power levels available in the downstream direction, the downstream data rates are adversely impacted as the cabMode parameter value is increased. Upstream data rates are not impacted since Cabinet Mode operation does not reduce the power levels used for upstream data transmission.

The rates shown in [Table 160](#) and [Table 161](#) are theoretical, maximum attainable rates on very short, clean loops. Actual rates in real work loops likely will not achieve the rates shown as loop plant conditions are less than ideal.

Table 160: MXK ADSL Annex A/M Transmit Filter Settings

Filter	Up Speed	Down Speed	Cutoff Frequency (Khz)
0 (off)	1029000	28115000	None

Table 160: MXK ADSL Annex A/M Transmit Filter Settings (Continued)

Filter	Up Speed	Down Speed	Cutoff Frequency (Khz)
1	1029000	22563000	603.75
2	1029000	21915000	646.875
3	1029000	21296000	690
4	1029000	20725000	733.125
5	1029000	20012000	776.25
6	1029000	19445000	819.375
7	1029000	18865000	862.5
8	1029000	18273000	905.625
9	1029000	17610000	948.75
10	1029000	17037000	991.875
11	1029000	16496000	1035
12	1029000	15861000	1078.125
13	1029000	15294000	1121.25
14	1029000	14693000	1164.375
15	1029000	14096000	1207.5



Tip: For MXK ADSL cards, the cut-off frequency shown in [Table 160](#) and [Table 161](#) represent the point in the PSD spectrum at which the power level will begin to be reduced, and can be estimated using the formula:

$$\text{Cut-off Freq} = [130 + 10 * (\text{cabMode})] * 4.3125 \text{ kHz}$$

Since there is a finite slope to the rate at which power can be reduced from the cut-off frequency, operators may need to select a **cabMode** parameter value greater than that shown in [Table 160](#) or [Table 161](#) to ensure that *all* power is eliminated below the target cut-off frequency.

Table 161: MXK ADSL Annex B Transmit Filter Settings

Filter	Up Speed	Down Speed	Cutoff Frequency (Khz)
0 (off)	1029000	27333000	None
1	1029000	22591000	603.75
2	1029000	21890000	646.875
3	1029000	21280000	690
4	1029000	20666000	733.125

Table 161: MXK ADSL Annex B Transmit Filter Settings (Continued)

Filter	Up Speed	Down Speed	Cutoff Frequency (Khz)
5	1029000	20086000	776.25
6	1029000	19489000	819.375
7	1029000	18875000	862.5
8	1029000	18277000	905.625
9	1029000	17610000	948.75
10	1029000	17031000	991.875
11	1029000	16451000	1035
12	1029000	15841000	1078.125
13	1029000	15258000	1121.25
14	1029000	14678000	1164.375
15	1029000	14064000	1207.5

To configure cabinet mode

- 1 Create/update the **adsl-co-profile** for the interface
- 2 Change the **cabMode** parameter.

The cabMode parameter will accept any number in the range from 0 - 15, the filter number defines the cutoff frequency.

```
zSH> update adsl-co-profile 1/1/1
```

```
adsl-co-profile 1/1/1
rateMode: -----> {adaptatruntime}
rateChanRatio: -----> {50}
targetSnrMgn: -----> {60}
maxSnrMgn: -----> {310}
minSnrMgn: -----> {0}
downshiftSnrMgn: -----> {0}
upshiftSnrMgn: -----> {0}
minUpshiftTime: -----> {0}
minDownshiftTime: -----> {0}
fastMinTxRate: -----> {32000}
interleaveMinTxRate: -----> {32000}
fastMaxTxRate: -----> {20000000}
maxInterleaveDelay: -----> {63}
interleaveMaxTxRate: -----> {20000000}
thresh15MinLofs: -----> {0}
thresh15MinLoss: -----> {0}
thresh15MinLols: -----> {0}
thresh15MinLprs: -----> {0}
thresh15MinESs: -----> {0}
threshFastRateUp: -----> {0}
```

```

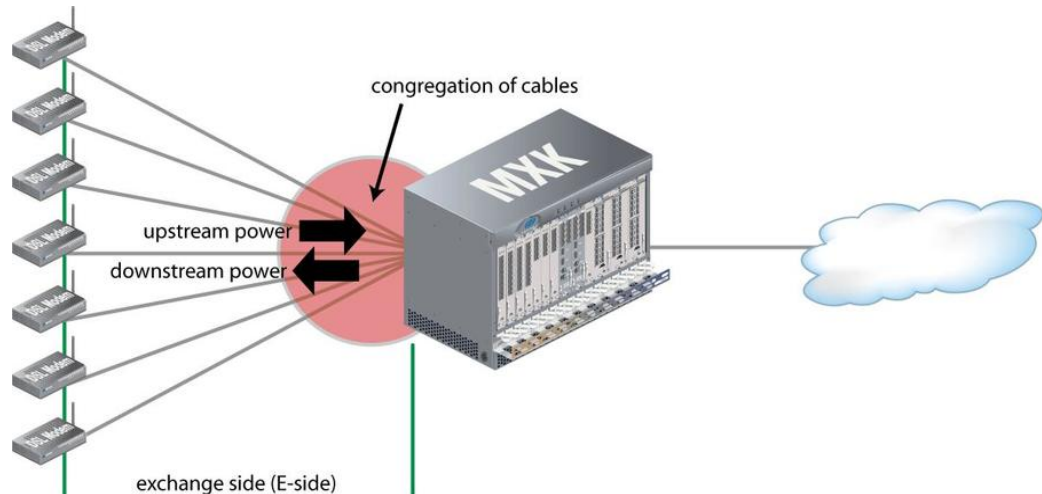
threshInterleaveRateUp: ---> {0}
threshFastRateDown: -----> {0}
threshInterleaveRateDown: -> {0}
initFailureTrapEnable: ----> {disable}
reachExtendedAdsl2: -----> {enable}
minTxThresholdRateAlarm: --> {0}
minINP: -----> {1}
phyRSupport: -----> {disable}
phyRmaxINP: -----> {0}
phyRminRSoverhead: -----> {0}
phyRRtxRatio: -----> {0}
txPowerAttenuation: -----> {0}
cabMode: -----> {0}:1 ==> 0 is disabled,
1 - 15 is enabled
.....
Save new record? [s]ave, [c]hange or [q]uit: s
Updated record saved.

```

Downstream Power Backoff (DPBO)

The MXK supports shaped downstream power backoff (DPBO) as described in ITU-T G.997. Like upstream power backoff, the idea of DPBO is to limit the interference generated where the cable congregates at the central office or street cabinet while still providing enough power to properly receive data from the far end device.

Figure 159: Both upstream and downstream power backoff reduce the power and hence the interference where the cables congregate at the CO or cabinet



DPBO is supported on the ADSL2+ and VDSL2 line cards.

For configuration information, please see [Downstream Power Backoff \(DPBO\)](#) on page 1106.

ADSL2+ cable and port pinouts

This section describes the ADSL2+ cables available from Zhone Technologies and the ADSL2+ port pinouts for both the 48-port ADSL2+ cards and the 72-port ADSL2+ card.

- [ADSL2+ bond 48-port card pinouts, page 1299](#)
- [ADSL2+ bond 48-port card cable pinouts, page 1303](#)
- [ADSL2+ bond 72-port card pinouts, page 1310](#)
- [ADSL2+ bond 72-port card cable pinouts, page 1315](#)

ADSL2+ bond 48-port card pinouts

This section describes the following ADSL2+ port pinouts.

[Table 162](#) lists the ADSL-48 card pinouts.

Table 162: ADSL-48 card pinouts

Port	Signal	Pin
1	Tip	J7-2
	Ring	J7-1
2	Tip	J7-4
	Ring	J7-3
3	Tip	J7-6
	Ring	J7-5
4	Tip	J7-8
	Ring	J7-7
5	Tip	J7-10
	Ring	J7-9
6	Tip	J7-12
	Ring	J7-11
7	Tip	J7-14
	Ring	J7-13
8	Tip	J7-16
	Ring	J7-15
9	Tip	J7-18
	Ring	J7-17
10	Tip	J7-20
	Ring	J7-19
11	Tip	J7-22
	Ring	J7-21
12	Tip	J7-24
	Ring	J7-23
13	Tip	J7-26
	Ring	J7-25
14	Tip	J7-28
	Ring	J7-27
15	Tip	J7-30
	Ring	J7-29

Table 162: ADSL-48 card pinouts (Continued)

Port	Signal	Pin
16	Tip	J7-32
	Ring	J7-31
17	Tip	J7-34
	Ring	J7-33
18	Tip	J7-36
	Ring	J7-35
19	Tip	J7-38
	Ring	J7-37
20	Tip	J7-40
	Ring	J7-39
21	Tip	J7-42
	Ring	J7-41
22	Tip	J7-44
	Ring	J7-43
23	Tip	J7-46
	Ring	J7-45
24	Tip	J7-48
	Ring	J7-47
25	Tip	J7-50
	Ring	J7-49
26	Tip	J7-52
	Ring	J7-51
27	Tip	J7-54
	Ring	J7-53
28	Tip	J7-56
	Ring	J7-55
29	Tip	J7-58
	Ring	J7-57
30	Tip	J7-60
	Ring	J7-59

Table 162: ADSL-48 card pinouts (Continued)

Port	Signal	Pin
31	Tip	J7-62
	Ring	J7-61
32	Tip	J7-64
	Ring	J7-63
33	Tip	J7-66
	Ring	J7-65
34	Tip	J7-68
	Ring	J7-67
35	Tip	J7-70
	Ring	J7-69
36	Tip	J7-72
	Ring	J7-71
37	Tip	J7-74
	Ring	J7-73
38	Tip	J7-76
	Ring	J7-75
39	Tip	J7-78
	Ring	J7-77
40	Tip	J7-80
	Ring	J7-79
41	Tip	J7-82
	Ring	J7-81
42	Tip	J7-84
	Ring	J7-83
43	Tip	J7-86
	Ring	J7-85
44	Tip	J7-88
	Ring	J7-87
45	Tip	J7-90
	Ring	J7-89

Table 162: ADSL-48 card pinouts (Continued)

Port	Signal	Pin
46	Tip	J7-92
	Ring	J7-91
47	Tip	J7-94
	Ring	J7-93
48	Tip	J7-96
	Ring	J7-95

ADSL2+ bond 48-port card cable pinouts

This section provides the following information:

- [ADSL-48 to dual 50-pin connector cable, page 1303](#)
- [ADSL 48-port card to dual 50-pin connector cables, page 1308](#)
- [Variations of ADSL2+ bond 48-port to dual 50-pin connector cables, page 1309](#)

ADSL-48 to dual 50-pin connector cable

[Figure 160](#) shows the ADSL-48 to dual 50-pin connector cable (MALC-CBL-ADSL-48). [Table 163 on page 1305](#) lists the ADSL-48 card pinouts. [Table 164 on page 1309](#) lists additional ADSL-48 to dual 50-pin connector cables. [Table 165 on page 1309](#) lists variations of these cables.

Figure 160: 48-port ADSL2+ to dual 50-pin cable

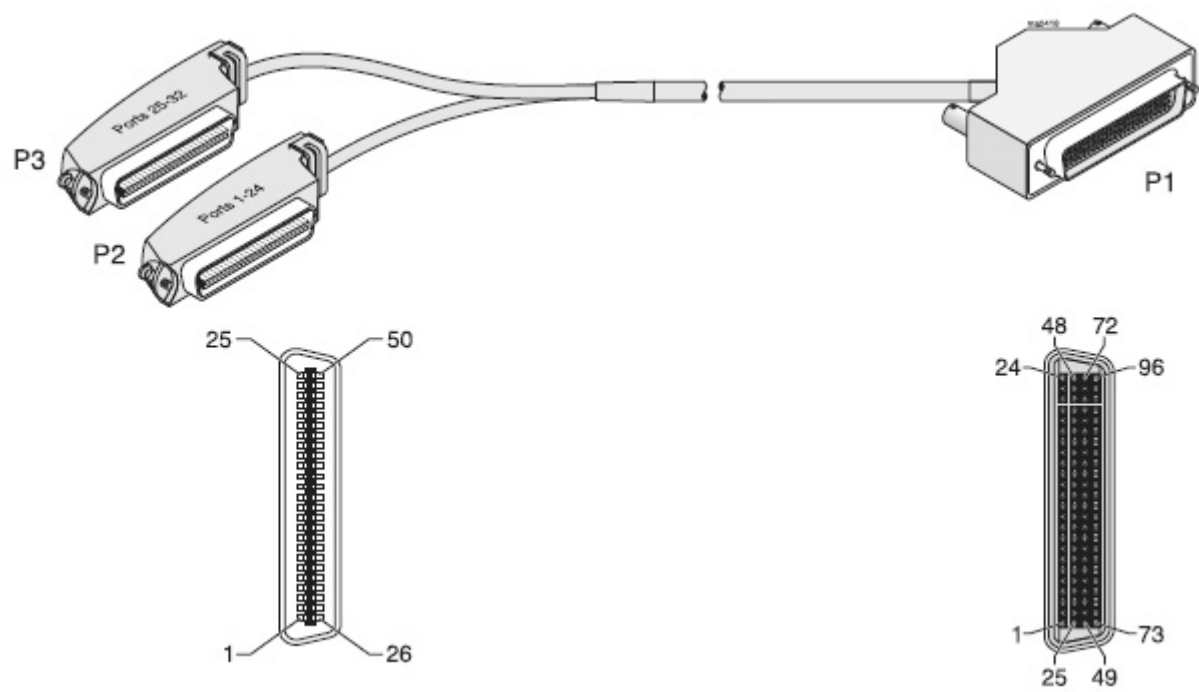


Table 163: 48-port ADSL to dual-50-pin cable pinouts

Pair	Signal	Color	From	To	Binder group
1	Tip	White/Blue	P1-2	P2-26	1 (Blue)
	Ring	Blue/White	P1-1	P2-1	
2	Tip	White/Orange	P1-4	P2-27	
	Ring	Orange/White	P1-3	P2-2	
3	Tip	White/Green	P1-6	P2-28	
	Ring	Green/White	P1-5	P2-3	
4	Tip	White/Brown	P1-8	P2-29	
	Ring	Brown/White	P1-7	P2-4	
5	Tip	White/Slate	P1-10	P2-30	
	Ring	Slate/White	P1-9	P2-5	
6	Tip	Red/Blue	P1-12	P2-31	
	Ring	Blue/Red	P1-11	P2-6	
7	Tip	Red/Orange	P1-14	P2-32	
	Ring	Orange/Red	P1-13	P2-7	
8	Tip	Red/Green	P1-16	P2-33	
	Ring	Green/Red	P1-15	P2-8	
9	Tip	Red/Brown	P1-18	P2-34	
	Ring	Brown/Red	P1-17	P2-9	
10	Tip	Red/Slate	P1-20	P2-35	
	Ring	Slate/Red	P1-19	P2-10	
11	Tip	Black/Blue	P1-22	P2-36	
	Ring	Blue/Black	P1-21	P2-11	
12	Tip	Black/Orange	P1-24	P2-37	
	Ring	Orange/Black	P1-23	P2-12	

Table 163: 48-port ADSL to dual-50-pin cable pinouts (Continued)

Pair	Signal	Color	From	To	Binder group
13	Tip	White/Blue	P1-26	P2-38	2 (Orange)
	Ring	Blue/White	P1-25	P2-13	
14	Tip	White/Orange	P1-28	P2-39	
	Ring	Orange/White	P1-27	P2-14	
15	Tip	White/Green	P1-30	P2-40	
	Ring	Green/White	P1-29	P2-15	
16	Tip	White/Brown	P1-32	P2-41	
	Ring	Brown/White	P1-31	P2-16	
17	Tip	White/Slate	P1-34	P2-42	
	Ring	Slate/White	P1-33	P2-17	
18	Tip	Red/Blue	P1-36	P2-43	
	Ring	Blue/Red	P1-35	P2-18	
19	Tip	Red/Orange	P1-38	P2-44	
	Ring	Orange/Red	P1-37	P2-19	
20	Tip	Red/Green	P1-40	P2-45	
	Ring	Green/Red	P1-39	P2-20	
21	Tip	Red/Brown	P1-42	P2-46	
	Ring	Brown/Red	P1-41	P2-21	
22	Tip	Red/Slate	P1-44	P2-47	
	Ring	Slate/Red	P1-43	P2-22	
23	Tip	Black/Blue	P1-46	P2-48	
	Ring	Blue/Black	P1-45	P2-23	
24	Tip	Black/Orange	P1-48	P2-49	
	Ring	Orange/Black	P1-47	P2-24	

Table 163: 48-port ADSL to dual-50-pin cable pinouts (Continued)

Pair	Signal	Color	From	To	Binder group
25	Tip	White/Blue	P1-50	P3-26	3 (Green)
	Ring	Blue/White	P1-49	P3-1	
26	Tip	White/Orange	P1-52	P3-27	
	Ring	Orange/White	P1-51	P3-2	
27	Tip	White/Green	P1-54	P3-28	
	Ring	Green/White	P1-53	P3-3	
28	Tip	White/Brown	P1-56	P3-29	
	Ring	Brown/White	P1-55	P3-4	
29	Tip	White/Slate	P1-58	P3-30	
	Ring	Slate/White	P1-57	P3-5	
30	Tip	Red/Blue	P1-60	P3-31	
	Ring	Blue/Red	P1-59	P3-6	
31	Tip	Red/Orange	P1-62	P3-32	
	Ring	Orange/Red	P1-61	P3-7	
32	Tip	Red/Green	P1-64	P3-33	
	Ring	Green/Red	P1-63	P3-8	
33	Tip	Red/Brown	P1-66	P3-34	
	Ring	Brown/Red	P1-65	P3-9	
34	Tip	Red/Slate	P1-68	P3-35	
	Ring	Slate/Red	P1-67	P3-10	
35	Tip	Black/Blue	P1-70	P3-36	
	Ring	Blue/Black	P1-69	P3-11	
36	Tip	Black/Orange	P1-72	P3-37	
	Ring	Orange/Black	P1-71	P3-12	

Table 163: 48-port ADSL to dual-50-pin cable pinouts (Continued)

Pair	Signal	Color	From	To	Binder group
37	Tip	White/Blue	P1-74	P3-38	4 (Brown)
	Ring	Blue/White	P1-73	P3-13	
38	Tip	White/Orange	P1-76	P3-39	
	Ring	Orange/White	P1-75	P3-14	
39	Tip	White/Green	P1-78	P3-40	
	Ring	Green/White	P1-77	P3-15	
40	Tip	White/Brown	P1-80	P3-41	
	Ring	Brown/White	P1-79	P3-16	
41	Tip	White/Slate	P1-82	P3-42	
	Ring	Slate/White	P1-81	P3-17	
42	Tip	Red/Blue	P1-84	P3-43	
	Ring	Blue/Red	P1-83	P3-18	
43	Tip	Red/Orange	P1-86	P3-44	
	Ring	Orange/Red	P1-85	P3-19	
44	Tip	Red/Green	P1-88	P3-45	
	Ring	Green/Red	P1-87	P3-20	
45	Tip	Red/Brown	P1-90	P3-46	
	Ring	Brown/Red	P1-89	P3-21	
46	Tip	Red/Slate	P1-92	P3-47	
	Ring	Slate/Red	P1-91	P3-22	
47	Tip	Black/Blue	P1-94	P3-48	
	Ring	Blue/Black	P1-93	P3-23	
48	Tip	Black/Orange	P1-96	P3-49	
	Ring	Orange/Black	P1-95	P3-24	

ADSL 48-port card to dual 50-pin connector cables

Table 164 describes the 48-port ADSL2+ to dual 50-pin connector cables.

Table 164: 48-port ADSL to dual 50-pin connector cables

MXK CABLE PART NAME	DESCRIPTION
MALC-CBL-ADSL-48	96 PIN TO (2) 50-PIN CONNECTOR, 48-PORT CARDS, 10 FT/ 3.05M
MALC-CBL-ADSL-48-10FTF	96PIN TO 2 50PIN FEMALE CONNECTOR, 10FT/3.05M
MALC-CBL-ADSL-48-15FT	96 PIN TO (2) 50-PIN CONNECTOR, 48-PORT CARDS, 15 FT/ 4.57M
MALC-CBL-ADSL-48-30M	96 PIN TO (2) 50-PIN CONNECTOR, 48-PORT CARDS, 98.4 FT/ 30M
MALC-CBL-ADSL-48-30MF	96 PIN TO (2) 50-PIN FEMALE CONNECTOR, 98.4FT/30M LENGTH
MALC-CBL-ADSL-48-4FT	96 PIN TO (2) 50-PIN CONNECTOR, 48-PORT CARDS, 4 FT/1.22M
MALC-CBL-ADSL-48-50M	96 PIN TO (2) 50-PIN CONNECTOR, 48-PORT CARDS, 164 FT/50M
MALC-CBL-ADSL-48-50FTF	96PIN TO 92) 50PIN CONNECTOR, 48-PORT CARDS, 50FT/ 15.24M
MALC-CBL-ADSL-48-70M	96 PIN TO (2) 50-PIN CONNECTOR, 48-PORT CARDS, 229 FT/70M
MALC-CBL-ADSL-48-125FTF	96PIN TO (2) 50PIN CONNECTOR, 48-PORT CARDS, 125FT/38.1M
MALC-CBL-ADSL-48-90DEG-10FT	96 PIN TO (2) 50-PIN CONNECTOR, 48-PORT CARDS, 90 DEGREES, 10 FT/3.05M
MALC-CBL-ADSL-48-90DEG-4FT	96 PIN TO (2) 50-PIN CONNECTOR, 48-PORT CARDS, 90 DEGREES, 4 FT/1.22M
MALC-CBL-ADSL-48-UP-30FT	96 PIN TO (2) 50-PIN CONNECTOR, 48-PORT CARDS, ROUTES UPWARDS, 30 FT/9.1M
MALC-CBL-ADSL-48-UP-60FT	96 PIN TO (2) 50-PIN CONNECTOR, 48-PORT CARDS, ROUTES UPWARDS, 60 FT/18.28M

Variations of ADSL2+ bond 48-port to dual 50-pin connector cables

[Table 165](#) lists variations of the 48-port ADSL2+ to dual 50-pin connector cables. These cables use the pinouts listed in [Table 163 on page 1305](#).

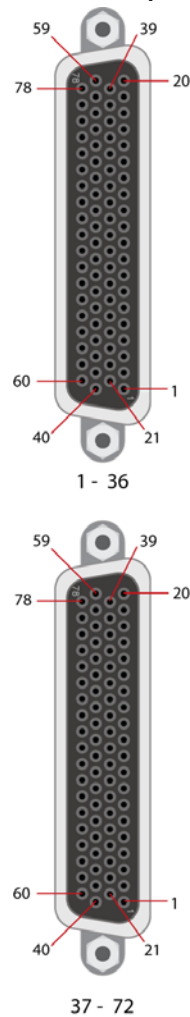
Table 165: Variations of 48-port ADSL to dual 50-pin connector cables

MXK CABLE PART NAME	DESCRIPTION
MALC-CBL-ADSL-48-100FT-BLUNT	96 PIN TO BLUNT END, 100 FT/30.5M
MALC-CBL-ADSL-48-350FT-BLUNT	96 PIN TO BLUNT END, 350 FT/106.7M

ADSL2+ bond 72-port card pinouts

The MXK-ADSL2+-BCM-72A card uses two standard HDD78 connectors. [Figure 161](#) shows the location of pinouts on the card connectors.

Figure 161: ADSL2+ bond 72-port card connector pinouts



[Table 166](#) lists the card pinouts of the top connector.

Table 166: 72-port POTS card pinouts - the top connector

Port	Signal	Pin
1	Tip	71
	Ring	70
2	Tip	73
	Ring	72

Table 166: 72-port POTS card pinouts - the top connector

Port	Signal	Pin
3	Tip	75
	Ring	74
4	Tip	77
	Ring	76
5	Tip	37
	Ring	38
6	Tip	35
	Ring	36
7	Tip	33
	Ring	34
8	Tip	31
	Ring	32
9	Tip	53
	Ring	52
10	Tip	55
	Ring	54
11	Tip	57
	Ring	56
12	Tip	59
	Ring	58
13	Tip	19
	Ring	20
14	Tip	17
	Ring	18
15	Tip	15
	Ring	16
16	Tip	13
	Ring	14
17	Tip	11
	Ring	12

Table 166: 72-port POTS card pinouts - the top connector

Port	Signal	Pin
18	Tip	9
	Ring	10
19	Tip	7
	Ring	8
20	Tip	29
	Ring	30
21	Tip	27
	Ring	28
22	Tip	25
	Ring	26
23	Tip	23
	Ring	24
24	Tip	21
	Ring	22
25	Tip	5
	Ring	6
26	Tip	3
	Ring	4
27	Tip	1
	Ring	2
28	Tip	41
	Ring	40
29	Tip	67
	Ring	66
30	Tip	65
	Ring	64
31	Tip	63
	Ring	62
32	Tip	61
	Ring	60

Table 166: 72-port POTS card pinouts - the top connector

Port	Signal	Pin
33	Tip	43
	Ring	42
34	Tip	45
	Ring	44
35	Tip	47
	Ring	46
36	Tip	49
	Ring	48

[Table 167](#) lists the card pinouts of the bottom connector.

Table 167: 72-port POTS card pinouts - the bottom connector

Port	Signal	Pin
37	Tip	71
	Ring	70
38	Tip	73
	Ring	72
39	Tip	75
	Ring	74
40	Tip	77
	Ring	76
41	Tip	37
	Ring	38
42	Tip	35
	Ring	36
43	Tip	33
	Ring	34
44	Tip	31
	Ring	32
45	Tip	53
	Ring	52

Table 167: 72-port POTS card pinouts - the bottom connector

Port	Signal	Pin
46	Tip	55
	Ring	54
47	Tip	57
	Ring	56
48	Tip	59
	Ring	58
49	Tip	13
	Ring	14
50	Tip	15
	Ring	16
51	Tip	17
	Ring	18
52	Tip	19
	Ring	20
53	Tip	21
	Ring	22
54	Tip	23
	Ring	24
55	Tip	25
	Ring	26
56	Tip	27
	Ring	28
57	Tip	11
	Ring	12
58	Tip	9
	Ring	10
59	Tip	7
	Ring	8
60	Tip	5
	Ring	6

Table 167: 72-port POTS card pinouts - the bottom connector

Port	Signal	Pin
61	Tip	3
	Ring	4
62	Tip	1
	Ring	2
63	Tip	29
	Ring	30
64	Tip	61
	Ring	60
65	Tip	63
	Ring	62
66	Tip	65
	Ring	64
67	Tip	67
	Ring	66
68	Tip	41
	Ring	40
69	Tip	43
	Ring	42
70	Tip	45
	Ring	44
71	Tip	47
	Ring	46
72	Tip	49
	Ring	48

ADSL2+ bond 72-port card cable pinouts

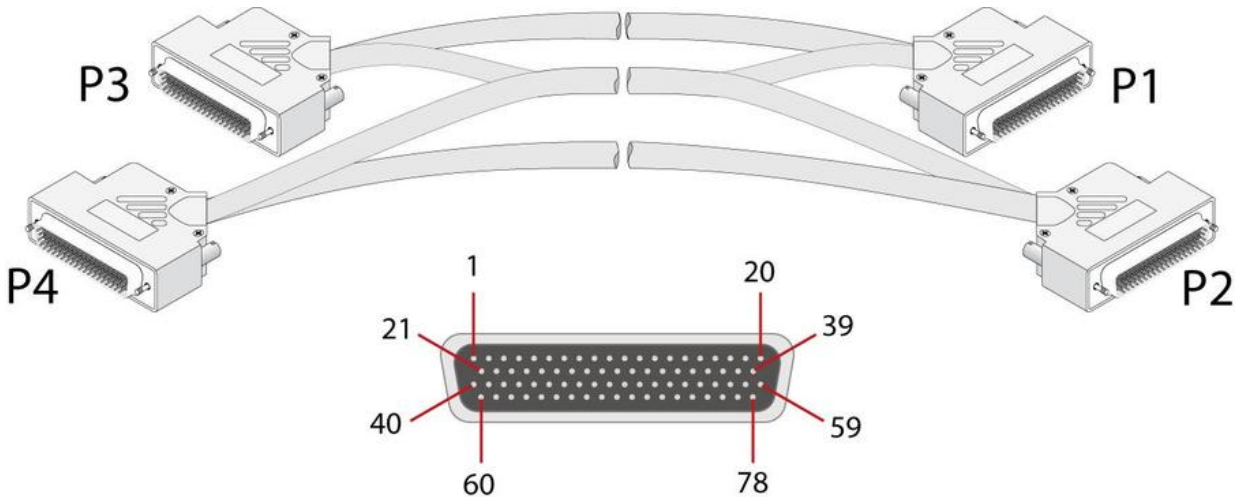
This section provides the following information:

- [dual 78-pin to dual 78-pin connector cable, page 1316](#)
- [ADSL-48 to dual 50-pin connector cable, page 1303](#)
- [dual 78-pin to blunt connector cable, page 1331](#)

dual 78-pin to dual 78-pin connector cable

Figure 162 shows the dual 78-pin to dual 78-pin connector cable used for 72-port POTS card. Table 164 on page 1309 lists variations of these cables. Table 169 on page 1318 lists pinouts of these cables.

Figure 162: dual 78-pin to dual 78-pin cable



Variations of dual 78-pin to dual 78-pin connector cables

Table 164 describes the dual 78-pin to dual 78-pin connector cables.

Table 168: dual 78-pin to dual 78-pin connector cables

MXK CABLE PART NAME	DESCRIPTION
MXK-CBL-72-72-CAT3-4FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 4FT/1.22M
MXK-CBL-72-72-CAT3-10FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 10FT/3.05M
MXK-CBL-72-72-CAT3-15FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 15FT/4.57M
MXK-CBL-72-72-CAT3-30FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 30FT/9.10M
MXK-CBL-72-72-CAT3-50FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 50FT/15.23M
MXK-CBL-72-72-CAT3-100FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 100FT/30.47M
MXK-CBL-72-72-CAT5-4FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 4FT/1.22M

Table 168: dual 78-pin to dual 78-pin connector cables (Continued)

MXK CABLE PART NAME	DESCRIPTION
MXK-CBL-72-72-CAT5-10FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 10FT/3.05M
MXK-CBL-72-72-CAT5-15FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 15FT/4.57M
MXK-CBL-72-72-CAT5-30FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 30FT/9.10M
MXK-CBL-72-72-CAT5-50FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 50FT/15.23M
MXK-CBL-72-72-CAT5-100FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 100FT/30.47M

dual 78-pin to dual 78-pin cable pinouts

[Table 169](#) lists pinouts of the dual 78-pin to dual 78-pin cables.

Table 169: dual 78-pin to dual 78-pin cable pinouts

Pair	Signal	Color	From	To	Cable
1	Tip	White/Blue	P1-71	P3-71	Cable 1
	Ring	Blue/White	P1-70	P3-70	
2	Tip	White/Orange	P1-73	P3-73	
	Ring	Orange/White	P1-72	P3-72	
3	Tip	White/Green	P1-75	P3-75	
	Ring	Green/White	P1-74	P3-74	
4	Tip	White/Brown	P1-77	P3-77	
	Ring	Brown/White	P1-76	P3-76	
5	Tip	White/Slate	P1-37	P3-37	
	Ring	Slate/White	P1-38	P3-38	
6	Tip	Red/Blue	P1-35	P3-35	
	Ring	Blue/Red	P1-36	P3-36	
7	Tip	Red/Orange	P1-33	P3-33	
	Ring	Orange/Red	P1-34	P3-34	
8	Tip	Red/Green	P1-31	P3-31	
	Ring	Green/Red	P1-32	P3-32	
9	Tip	Red/Brown	P1-53	P3-53	
	Ring	Brown/Red	P1-52	P3-52	
10	Tip	Red/Slate	P1-55	P3-55	
	Ring	Slate/Red	P1-54	P3-54	
11	Tip	Black/Blue	P1-57	P3-57	
	Ring	Blue/Black	P1-56	P3-56	
12	Tip	Black/Orange	P1-59	P3-59	
	Ring	Orange/Black	P1-58	P3-58	
13	Tip	Black/Green	P1-19	P3-19	
	Ring	Green/Black	P1-20	P3-20	
14	Tip	Black/Brown	P1-17	P3-17	
	Ring	Brown/Black	P1-18	P3-18	

Table 169: dual 78-pin to dual 78-pin cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
15	Tip	Black/Slate	P1-15	P3-15	Cable 1 (Continued)
	Ring	Slate/Black	P1-16	P3-16	
16	Tip	Yellow/Blue	P1-13	P3-13	
	Ring	Blue/Yellow	P1-14	P3-14	
17	Tip	Yellow/Orange	P1-11	P3-11	
	Ring	Orange/Yellow	P1-12	P3-12	
18	Tip	Yellow/Green	P1-9	P3-9	
	Ring	Green/Yellow	P1-10	P3-10	
19	Tip	Yellow/Brown	P1-7	P3-7	
	Ring	Brown/Yellow	P1-8	P3-8	
20	Tip	Yellow/Slate	P1-29	P3-29	
	Ring	Slate/Yellow	P1-30	P3-30	
21	Tip	Violet/Blue	P1-27	P3-27	
	Ring	Blue/Violet	P1-28	P3-28	
22	Tip	Violet/Orange	P1-25	P3-25	
	Ring	Orange/Violet	P1-26	P3-26	
23	Tip	Violet/Green	P1-23	P3-23	
	Ring	Green/Violet	P1-24	P3-24	
24	Tip	Violet/Brown	P1-21	P3-21	
	Ring	Brown/Violet	P1-22	P3-22	

Table 169: dual 78-pin to dual 78-pin cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
25	Tip	White/Blue	P1-5	P3-5	Cable 2
	Ring	Blue/White	P1-6	P3-6	
26	Tip	White/Orange	P1-3	P3-3	
	Ring	Orange/White	P1-4	P3-4	
27	Tip	White/Green	P1-1	P3-1	
	Ring	Green/White	P1-2	P3-2	
28	Tip	White/Brown	P1-41	P3-41	
	Ring	Brown/White	P1-40	P3-40	
29	Tip	White/Slate	P1-67	P3-67	
	Ring	Slate/White	P1-66	P3-66	
30	Tip	Red/Blue	P1-65	P3-65	
	Ring	Blue/Red	P1-64	P3-64	
31	Tip	Red/Orange	P1-63	P3-63	
	Ring	Orange/Red	P1-62	P3-62	
32	Tip	Red/Green	P1-61	P3-61	
	Ring	Green/Red	P1-60	P3-60	
33	Tip	Red/Brown	P1-43	P3-43	
	Ring	Brown/Red	P1-42	P3-42	
34	Tip	Red/Slate	P1-45	P3-45	
	Ring	Slate/Red	P1-44	P3-44	
35	Tip	Black/Blue	P1-47	P3-47	
	Ring	Blue/Black	P1-46	P3-46	
36	Tip	Black/Orange	P1-49	P3-49	
	Ring	Orange/Black	P1-48	P3-48	

Table 169: dual 78-pin to dual 78-pin cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
37	Tip	Black/Green	P2-71	P4-71	Cable 2 (Continued)
	Ring	Green/Black	P2-70	P4-70	
38	Tip	Black/Brown	P2-73	P4-73	
	Ring	Brown/Black	P2-72	P4-72	
39	Tip	Black/Slate	P2-75	P4-75	
	Ring	Slate/Black	P2-74	P4-74	
40	Tip	Yellow/Blue	P2-77	P4-77	
	Ring	Blue/Yellow	P2-76	P4-76	
41	Tip	Yellow/Orange	P2-37	P4-37	
	Ring	Orange/Yellow	P2-38	P4-38	
42	Tip	Yellow/Green	P2-35	P4-35	
	Ring	Green/Yellow	P2-36	P4-36	
43	Tip	Yellow/Brown	P2-33	P4-33	
	Ring	Brown/Yellow	P2-34	P4-34	
44	Tip	Yellow/Slate	P2-31	P4-31	
	Ring	Slate/Yellow	P2-32	P4-32	
45	Tip	Violet/Blue	P2-53	P4-53	
	Ring	Blue/Violet	P2-52	P4-52	
46	Tip	Violet/Orange	P2-55	P4-55	
	Ring	Orange/Violet	P2-54	P4-54	
47	Tip	Violet/Green	P2-57	P4-57	
	Ring	Green/Violet	P2-56	P4-56	
48	Tip	Violet/Brown	P2-59	P4-59	
	Ring	Brown/Violet	P2-58	P4-58	

Table 169: dual 78-pin to dual 78-pin cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
49	Tip	White/Blue	P2-13	P4-13	Cable 3
	Ring	Blue/White	P2-14	P4-14	
50	Tip	White/Orange	P2-15	P4-15	
	Ring	Orange/White	P2-16	P4-16	
51	Tip	White/Green	P2-17	P4-17	
	Ring	Green/White	P2-18	P4-18	
52	Tip	White/Brown	P2-19	P4-19	
	Ring	Brown/White	P2-20	P4-20	
53	Tip	White/Slate	P2-21	P4-21	
	Ring	Slate/White	P2-22	P4-22	
54	Tip	Red/Blue	P2-23	P4-23	
	Ring	Blue/Red	P2-24	P4-24	
55	Tip	Red/Orange	P2-25	P4-25	
	Ring	Orange/Red	P2-26	P4-26	
56	Tip	Red/Green	P2-27	P4-27	
	Ring	Green/Red	P2-28	P4-28	
57	Tip	Red/Brown	P2-11	P4-11	
	Ring	Brown/Red	P2-12	P4-12	
58	Tip	Red/Slate	P2-9	P4-9	
	Ring	Slate/Red	P2-10	P4-10	
59	Tip	Black/Blue	P2-7	P4-7	
	Ring	Blue/Black	P2-8	P4-8	
60	Tip	Black/Orange	P2-5	P4-5	
	Ring	Orange/Black	P2-6	P4-6	
61	Tip	Black/Green	P2-3	P4-3	
	Ring	Green/Black	P2-4	P4-4	
62	Tip	Black/Brown	P2-1	P4-1	
	Ring	Brown/Black	P2-2	P4-2	
63	Tip	Black/Slate	P2-29	P4-29	
	Ring	slate/Black	P2-30	P4-30	

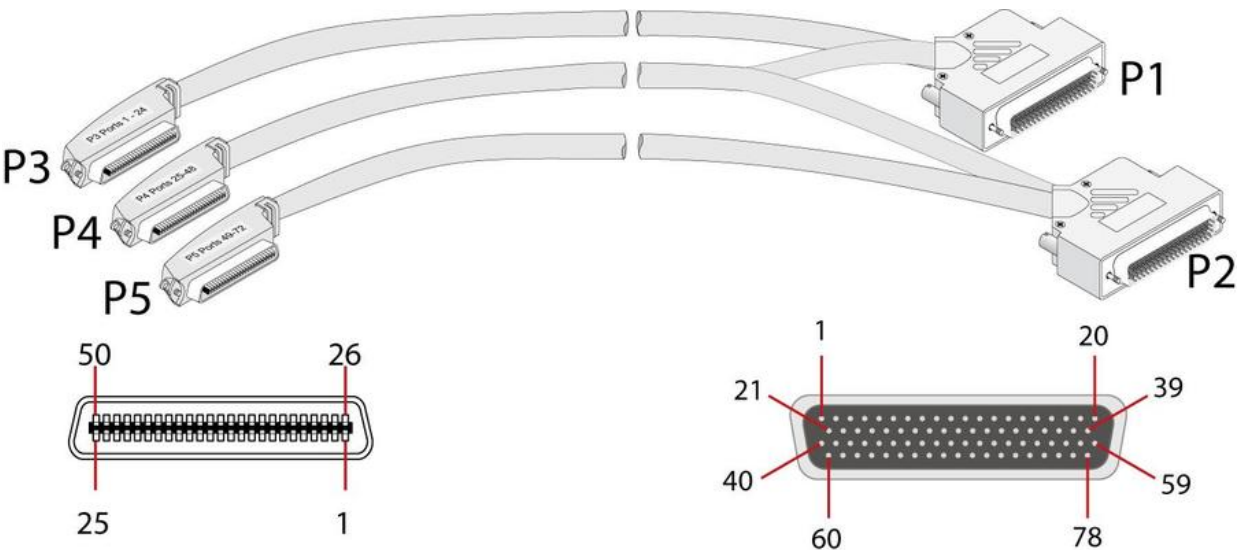
Table 169: dual 78-pin to dual 78-pin cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
64	Tip	Yellow/Blue	P2-61	P4-61	Cable 3 (Continued)
	Ring	Blue/Yellow	P2-60	P4-60	
65	Tip	Yellow/Orange	P2-63	P4-63	
	Ring	Orange/Yellow	P2-62	P4-62	
66	Tip	Yellow/Green	P2-65	P4-65	
	Ring	Green/Yellow	P2-64	P4-64	
67	Tip	Yellow/Brown	P2-67	P4-67	
	Ring	Brown/Yellow	P2-66	P4-66	
68	Tip	Yellow/Slate	P2-41	P4-41	
	Ring	Slate/Yellow	P2-40	P4-40	
69	Tip	Violet/Blue	P2-43	P4-43	
	Ring	Blue/Violet	P2-42	P4-42	
70	Tip	Violet/Orange	P2-45	P4-45	
	Ring	Orange/Violet	P2-44	P4-44	
71	Tip	Black/Orange	P2-47	P4-47	
	Ring	Orange/Black	P2-46	P4-46	
72	Tip	Violet/Brown	P2-49	P4-49	
	Ring	Brown/Violet	P2-48	P4-48	

dual 78-pin to three 50-pin connector cable

[Figure 160](#) shows the dual 78-pin to three 50-pin connectors cables for 72-port POTS card. [Table 165 on page 1309](#) lists variations of these cables. [Table 171 on page 1326](#) lists pinouts of these cables.

Figure 163: dual 78-pin to three 50-pin cable



Variations of dual 78-pin to three 50-pin connector cables

Table 165 lists variations of the dual 78-pin to three 50-pin connector cables.

Table 170: Variations of dual 78-pin to three 50-pin connector cables

MXK CABLE PART NAME	DESCRIPTION
MXK-CBL-72-CAT3-4FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 4FT/1.22M
MXK-CBL-72-CAT3-10FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 10FT/3.05M
MXK-CBL-72-CAT3-15FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 15FT/4.57M
MXK-CBL-72-CAT3-30FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 30FT/9.10M
MXK-CBL-72-CAT3-50FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 50FT/15.23M
MXK-CBL-72-CAT3-100FT	CABLE: (2) 78PIN TO (3) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 100FT/30.47M
MXK-CBL-72-CAT5-4FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 4FT/1.22M
MXK-CBL-72-CAT5-10FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 10FT/3.05M
MXK-CBL-72--CAT5-15FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 15FT/4.57M

Table 170: Variations of dual 78-pin to three 50-pin connector cables (Continued)

MXK CABLE PART NAME	DESCRIPTION
MXK-CBL-72-CAT5-30FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 30FT/9.10M
MXK-CBL-72-CAT5-50FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 50FT/15.23M
MXK-CBL-72-CAT5-100FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 100FT/30.47M

dual 78-pin to three 50-pin cable or blunt pinouts**Table 171: dual 78-pin to three 50-pins or blunt cable pinouts**

Pair	Signal	Color	From	To	Cable
1	Tip	White/Blue	P1-71	P3-26	Cable 1
	Ring	Blue/White	P1-70	P3-1	
2	Tip	White/Orange	P1-73	P3-27	
	Ring	Orange/White	P1-72	P3-2	
3	Tip	White/Green	P1-75	P3-28	
	Ring	Green/White	P1-74	P3-3	
4	Tip	White/Brown	P1-77	P3-29	
	Ring	Brown/White	P1-76	P3-4	
5	Tip	White/Slate	P1-37	P3-30	
	Ring	Slate/White	P1-38	P3-5	
6	Tip	Red/Blue	P1-35	P3-31	
	Ring	Blue/Red	P1-36	P3-6	
7	Tip	Red/Orange	P1-33	P3-32	
	Ring	Orange/Red	P1-34	P3-7	
8	Tip	Red/Green	P1-31	P3-33	
	Ring	Green/Red	P1-32	P3-8	
9	Tip	Red/Brown	P1-53	P3-34	
	Ring	Brown/Red	P1-52	P3-9	
10	Tip	Red/Slate	P1-55	P3-35	
	Ring	Slate/Red	P1-54	P3-10	
11	Tip	Black/Blue	P1-57	P3-36	
	Ring	Blue/Black	P1-56	P3-11	
12	Tip	Black/Orange	P1-59	P3-37	
	Ring	Orange/Black	P1-58	P3-12	
13	Tip	Black/Green	P1-19	P3-38	
	Ring	Green/Black	P1-20	P3-13	
14	Tip	Black/Brown	P1-17	P3-39	
	Ring	Brown/Black	P1-18	P3-14	

Table 171: dual 78-pin to three 50-pins or blunt cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
15	Tip	Black/Slate	P1-15	P3-40	Cable 1 (Continued)
	Ring	Slate/Black	P1-16	P3-15	
16	Tip	Yellow/Blue	P1-13	P3-41	
	Ring	Blue/Yellow	P1-14	P3-16	
17	Tip	Yellow/Orange	P1-11	P3-42	
	Ring	Orange/Yellow	P1-12	P3-17	
18	Tip	Yellow/Green	P1-9	P3-43	
	Ring	Green/Yellow	P1-10	P3-18	
19	Tip	Yellow/Brown	P1-7	P3-44	
	Ring	Brown/Yellow	P1-8	P3-19	
20	Tip	Yellow/Slate	P1-29	P3-45	
	Ring	Slate/Yellow	P1-30	P3-20	
21	Tip	Violet/Blue	P1-27	P3-46	
	Ring	Blue/Violet	P1-28	P3-21	
22	Tip	Violet/Orange	P1-25	P3-47	
	Ring	Orange/Violet	P1-26	P3-22	
23	Tip	Violet/Green	P1-23	P3-48	
	Ring	Green/Violet	P1-24	P3-23	
24	Tip	Violet/Brown	P1-21	P3-49	
	Ring	Brown/Violet	P1-22	P3-24	

Table 171: dual 78-pin to three 50-pins or blunt cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
25	Tip	White/Blue	P1-5	P4-26	Cable 2
	Ring	Blue/White	P1-6	P4-1	
26	Tip	White/Orange	P1-3	P4-27	
	Ring	Orange/White	P1-4	P4-2	
27	Tip	White/Green	P1-1	P4-28	
	Ring	Green/White	P1-2	P4-3	
28	Tip	White/Brown	P1-41	P4-29	
	Ring	Brown/White	P1-40	P4-4	
29	Tip	White/Slate	P1-67	P4-30	
	Ring	Slate/White	P1-66	P4-5	
30	Tip	Red/Blue	P1-65	P4-31	
	Ring	Blue/Red	P1-64	P4-6	
31	Tip	Red/Orange	P1-63	P4-32	
	Ring	Orange/Red	P1-62	P4-7	
32	Tip	Red/Green	P1-61	P4-33	
	Ring	Green/Red	P1-60	P4-8	
33	Tip	Red/Brown	P1-43	P4-34	
	Ring	Brown/Red	P1-42	P4-9	
34	Tip	Red/Slate	P1-45	P4-35	
	Ring	Slate/Red	P1-44	P4-10	
35	Tip	Black/Blue	P1-47	P4-36	
	Ring	Blue/Black	P1-46	P4-11	
36	Tip	Black/Orange	P1-49	P4-37	
	Ring	Orange/Black	P1-48	P4-12	

Table 171: dual 78-pin to three 50-pins or blunt cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
37	Tip	Black/Green	P2-71	P4-38	Cable 2 (Continued)
	Ring	Green/Black	P2-70	P4-13	
38	Tip	Black/Brown	P2-73	P4-39	
	Ring	Brown/Black	P2-72	P4-14	
39	Tip	Black/Slate	P2-75	P4-40	
	Ring	Slate/Black	P2-74	P4-15	
40	Tip	Yellow/Blue	P2-77	P4-41	
	Ring	Blue/Yellow	P2-76	P4-16	
41	Tip	Yellow/Orange	P2-37	P4-42	
	Ring	Orange/Yellow	P2-38	P4-17	
42	Tip	Yellow/Green	P2-35	P4-43	
	Ring	Green/Yellow	P2-36	P4-18	
43	Tip	Yellow/Brown	P2-33	P4-44	
	Ring	Brown/Yellow	P2-34	P4-19	
44	Tip	Yellow/Slate	P2-31	P4-45	
	Ring	Slate/Yellow	P2-32	P4-20	
45	Tip	Violet/Blue	P2-53	P4-46	
	Ring	Blue/Violet	P2-52	P4-21	
46	Tip	Violet/Orange	P2-55	P4-47	
	Ring	Orange/Violet	P2-54	P4-22	
47	Tip	Violet/Green	P2-57	P4-48	
	Ring	Green/Violet	P2-56	P4-23	
48	Tip	Violet/Brown	P2-59	P4-49	
	Ring	Brown/Violet	P2-58	P4-24	

Table 171: dual 78-pin to three 50-pins or blunt cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
49	Tip	White/Blue	P2-13	P5-26	Cable 3
	Ring	Blue/White	P2-14	P5-1	
50	Tip	White/Orange	P2-15	P5-27	
	Ring	Orange/White	P2-16	P5-2	
51	Tip	White/Green	P2-17	P5-28	
	Ring	Green/White	P2-18	P5-3	
52	Tip	White/Brown	P2-19	P5-29	
	Ring	Brown/White	P2-20	P5-4	
53	Tip	White/Slate	P2-21	P5-30	
	Ring	Slate/White	P2-22	P5-5	
54	Tip	Red/Blue	P2-23	P5-31	
	Ring	Blue/Red	P2-24	P5-6	
55	Tip	Red/Orange	P2-25	P5-32	
	Ring	Orange/Red	P2-26	P5-7	
56	Tip	Red/Green	P2-27	P5-33	
	Ring	Green/Red	P2-28	P5-8	
57	Tip	Red/Brown	P2-11	P5-34	
	Ring	Brown/Red	P2-12	P5-9	
58	Tip	Red/Slate	P2-9	P5-35	
	Ring	Slate/Red	P2-10	P5-10	
59	Tip	Black/Blue	P2-7	P5-36	
	Ring	Blue/Black	P2-8	P5-11	
60	Tip	Black/Orange	P2-5	P5-37	
	Ring	Orange/Black	P2-6	P5-12	
61	Tip	Black/Green	P2-3	P5-38	
	Ring	Green/Black	P2-4	P5-13	
62	Tip	Black/Brown	P2-1	P5-39	
	Ring	Brown/Black	P2-2	P5-14	
63	Tip	Black/Slate	P2-29	P5-40	
	Ring	slate/Black	P2-30	P5-15	

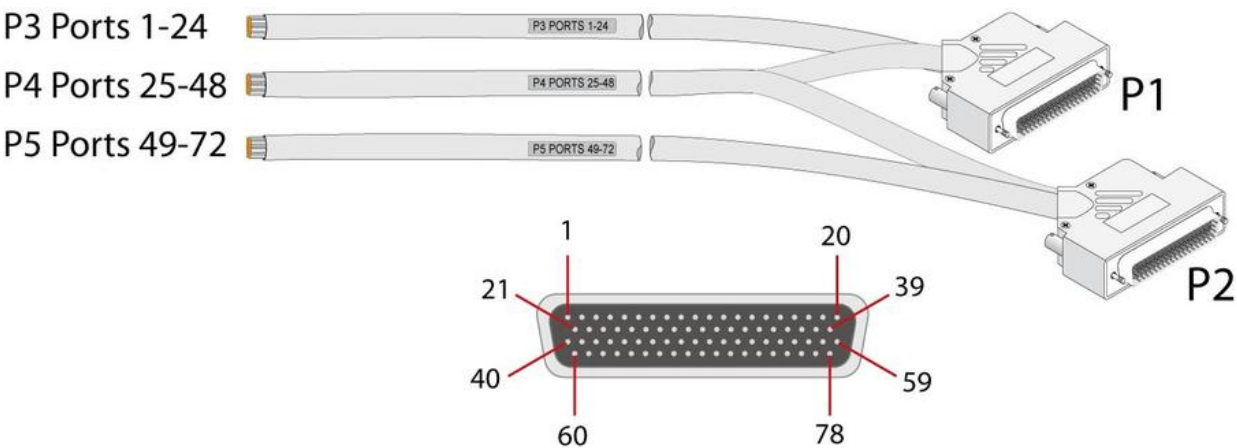
Table 171: dual 78-pin to three 50-pins or blunt cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
64	Tip	Yellow/Blue	P2-61	P5-41	Cable 3 (Continued)
	Ring	Blue/Yellow	P2-60	P5-16	
65	Tip	Yellow/Orange	P2-63	P5-42	
	Ring	Orange/Yellow	P2-62	P5-17	
66	Tip	Yellow/Green	P2-65	P5-43	
	Ring	Green/Yellow	P2-64	P5-18	
67	Tip	Yellow/Brown	P2-67	P5-44	
	Ring	Brown/Yellow	P2-66	P5-19	
68	Tip	Yellow/Slate	P2-41	P5-45	
	Ring	Slate/Yellow	P2-40	P5-20	
69	Tip	Violet/Blue	P2-43	P5-46	
	Ring	Blue/Violet	P2-42	P5-21	
70	Tip	Violet/Orange	P2-45	P5-47	
	Ring	Orange/Violet	P2-44	P5-22	
71	Tip	Black/Orange	P2-47	P5-48	
	Ring	Orange/Black	P2-46	P5-23	
72	Tip	Violet/Brown	P2-49	P5-49	
	Ring	Brown/Violet	P2-48	P5-24	

dual 78-pin to blunt connector cable

[Figure 164](#) shows the dual 78-pin to blunt connector cable for 72-port POTS card. [Table 172 on page 1332](#) lists variations of these cables. [Table 171 on page 1326](#) lists pinouts of these cables. Note that the pinouts of two to blunt are same as two to three cables.

Figure 164: dual 78-pin to blunt cable



Variations of dual 78-pin to blunt end cables

Table 172 lists variations of the dual 78-pin to blunt end cables for 72-port POTS card.

Table 172: Variations of dual 78-pin to blunt connector cables

MXK CABLE PART NAME	DESCRIPTION
MXK-CBL-72-CAT3-4FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT3 UNSHIELDED, 4FT/1.22M
MXK-CBL-72-CAT3-10FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT3 UNSHIELDED, 10FT/3.05M
MXK-CBL-72-CAT3-15FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT3 UNSHIELDED, 15FT/4.57M
MXK-CBL-72-CAT3-30FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT3 UNSHIELDED, 30FT/9.10M
MXK-CBL-72-CAT3-50FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT3 UNSHIELDED, 50FT/15.23M
MXK-CBL-72-CAT3-100FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT3 UNSHIELDED, 100FT/30.47M
MXK-CBL-72-CAT5-4FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT5 SHIELDED, 4FT/1.22M
MXK-CBL-72-CAT5-10FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT5 SHIELDED, 10FT/3.05M
MXK-CBL-72-CAT5-15FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT5 SHIELDED, 15FT/4.57M
MXK-CBL-72-CAT5-30FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT5 SHIELDED, 30FT/9.10M

Table 172: Variations of dual 78-pin to blunt connector cables (Continued)

MXK CABLE PART NAME	DESCRIPTION
MXK-CBL-72-CAT5-50FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT5 SHIELDED, 50FT/15.23M
MXK-CBL-72-CAT5-100FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT5 SHIELDED, 100FT/30.47M

ADSL2+ testing (SELT/DELT) on the MXK

This section discusses:

- [SELT \(Single-End Loop Test\), page 1334](#)
- [DELT \(Dual-End Loop Test\), page 1339](#)

Single-End Loop Test (SELT) and Dual-End Loop Test (DELT) are loop tests which can be used to proactively pre-qualify a loop (SELT) or reactively test a loop after a modem has been deployed (DELT).



Note: Only one test (SELT, DELT, or TAC) can run at a time on an interface. Otherwise the current test will be interrupted by the newly configured test, thus the test results returned will be inaccurate.

Before starting a SELT/ DELT test on an interface, you can use the **selt| delt show status <interface>** command to identify whether there is a SELT or DELT test is running on this interface or not.



Note: Only one SELT test can run at a time on a board. But multiple DELT tests can run at same time on a board, as long as they are not running in the same interface.



Note: Test limitations:

- Test range is 600 to 9000 feet.
- Mixed gauge wire is not supported.
- Results have +/- 10% variance.

SELT (Single-End Loop Test)

SELT is a single-ended test. A copper loop can be tested from the MXK only, without the need for any external test equipment in either the CO or at the remote end of the loop. SELT is primarily used for proactive loop pre-qualification.

Starting SELT

The SELT algorithm estimate length and gauge of the wire connected to the interface. It also measures loop noise floor at each ADSL2+ subcarrier frequency. There must be no CPE connected to the line, even if it is turned off, although a phone maybe connected. Only one SELT test maybe run at a time.

Use the `selt start <interface>` command to start a SELT test on an ADSL2+ interface. Make sure the administrative status on this ADSL2+ interface is down.

- 1 Set the administrative status of the ADSL2+ interface to down.

```
zSH> port down 1-4-3-0/adsl1
```

1-4-3-0/adsl set to admin state DOWN

2 Start the SELT test on it.

```
zSH> selt start 1-4-3-0/adsl
Selt test started on interface 1-4-3-0/adsl
```

Showing SELT status

Use the **selt show status <interface>** command to display SELT test progress and loop characterization on an ADSL2+ interface.

[Table 173](#) describes the SELT test status parameters.

Table 173: SELT test status parameters

Parameter	Description
status	The status of the current SELT test. Values: in-progress This SELT is currently running complete This SELT has been run at least once since system reset and has completed aborted This SELT was aborted stopped-in-progress The request to abort the test is still in progress not-started This SELT has not been run on this interface since last system reset results-cleared This SELT test results table was removed as requested. time-out This SELT was aborted automatically because it did not complete within the max-duration. unsupported The device or interface does not support SELT port-active The interface's administrative status has not been set to down as required. measurement invalid This SELT returns invalid measurements
max-duration	Maximum number of seconds the test is allowed to run. Default value is Disabled.
cfg-gauge	Configured expected wire gauge. Default value is awg 26.
cfg-cable	Configured expected cable type, real or simulated. Default value is real.
time-left	Estimated number of seconds remaining in the current test.
device	ADSL chip set in MXK used to perform SELT test.
bridge-taps	Whether or not this chip set can run SELT test in the presence of bridge taps.
date-time	Date and time that the most-recently run test completed.

Table 173: SELT test status parameters (Continued)

Parameter	Description
length	Estimated length of the loop.
gauge	Estimated wire gauge (if supported by hardware)

The following examples show the SELT status during the test and after the test completed:

```

zSH> selt show status 1-4-3-0/adsl
status:      in-progress
max-duration: disabled
cfg-gauge:   awg26
cfg-cable:   real
time-left:   307 seconds
device:      broadcom-6411
bridge-taps: not-supported
date-time:   no results have been generated
length:      unknown
gauge:       unknown

zSH> selt show status 1-4-3-0/adsl
status:      complete
max-duration: disabled
cfg-gauge:   awg26
cfg-cable:   real
time-left:   0 seconds
device:      broadcom-6411
bridge-taps: not-supported
date-time:   results generated 30 jun 2009, 22:35:32
length:      131 feet
gauge:       awg26

```

Showing SELT noise

Use the **selt show noise <interface> [start-index [num-vals]]** command to display the loop noise floor results of a SELT test. There is one noise measurement per ADSL2+ tone. Each noise value has units of dBm/Hz.

- <interface> can be in the form of ifIndex (e.g. 432), name/type (e.g. 1-4-1-0/adsl) or shelf/slot/port/subport/type (e.g. 1/4/1/0/adsl).
- [start-index]: (0..511) is the tone index with which to start. 0 is 4.3125 kHz, 1 is 8.625 kHz, up to 511 which is 2208.0000 kHz.
- [num-vals]: the number of tones to display.

```

zSH> selt show noise 1-4-3-0/adsl
Results generated 10 sep 2006, 14:35:56.
Tone   Tone Freq   Noise
Index  (kHz)        (dBm/Hz)
-----
      0      4.3125     -95.7
      1      8.6250     -118.3

```

```

2      12.9375    -121.4
3      17.2500    -123.8
4      21.5625    -124.9
5      25.8750    -126.3
6      30.1875    -125.5
7      34.5000    -121.8
8      38.8125    -113.6
9      43.1250    -125.9
10     47.4375    -127.7
11     51.7500    -128.4
12     56.0625    -128.3
13     60.3750    -128.5
14     64.6875    -128.3
15     69.0000    -124.4
[etc, up to index 511]

```

```

zSH> selt show noise 1-4-1-0/ads1 253 6
Results generated 10 sep 2006, 14:35:56.
Tone   Tone Freq   Noise
Index   (kHz)         (dBm/Hz)
-----
253    1095.3750     -122.0
254    1099.6875     -122.6
255    1104.0000     -121.9
256    1108.3125     no measurement
257    1112.6250     no measurement
258    1116.9375     no measurement
259    1121.2500     no measurement

```

Setting SELT

The MXK supports the following SELT commands to change the SELT settings.

- **selt set units <awg | metric | japan>**

Set the SELT display units for the chassis.

- awg: Show wire diameters and lengths in English units. This is default value.
- metric: Show wire diameters and lengths in Metric units.
- japan: Show wire diameters and lengths in Japanese metric units.

- **selt set max-duration <interface> <num-seconds>**

Set the maximum amount of time, in seconds, that a SELT test can run on the ADSL2+ interface. If a SELT test runs for more than <num-seconds> it will be aborted.

Setting max-duration to zero disables SELT timeout on an interface. By default, max-duration is Disabled.

Note that, in order to get valid results, a SELT test must be allowed to run to completion, and this may take several minutes, depending on the speed of the processor used to do the computation.

- **selt set gauge <interface> <wire-gauge>**

Set the expected diameter of the wire connected to an ADSL2+ interface. The diameter may be set using any units, regardless of the display units set with the **selt set units** command. The <wire-gauge> option must use one of these settings:

- unknown: unknown wire gauge
- awg19: 19 gauge
- awg22: 22 gauge
- awg24: 24 gauge
- awg26: 26 gauge. This is default value.
- 32mm: 0.32 millimeters
- 40mm - 0.40 millimeters
- 50mm - 0.50 millimeters
- 63mm - 0.63 millimeters
- 65mm - 0.65 millimeters
- 90mm - 0.90 millimeters

- **selt set cable <interface> <cable-type>**

Set the type of cable being tested, real or simulated.

<cable-type>: real, ds190, ds400. The *real* setting indicates that an actual physical cable is connected to the interface, and this is the default setting. In a lab or test environment, the cable may be simulated and use the *dsl90* or *dsl400* setting.

Examples:

```
zSH> selt set units metric
Selt information will be displayed in metric units.
```

```
zSH> selt set max-duration 1-4-3-0/adsl 200
Selt test timeout on interface 1-4-3-0/adsl set to 200 seconds.
```

Aborting SELT

Use the **selt abort <interface>** command to terminate a SELT test on an ADSL2+ interface. Note that it may take some time (perhaps as much as a minute) for the SELT test to abort.

```
zSH> selt abort 1-4-3-0/adsl
Selt test aborted on interface 1-4-3-0/adsl
```

Clearing stored DELT results

Use the **selt clear <interface>** command to clear stored results of a SELT on an ADSL2+ interface.

```
zSH> selt clear 1-4-3-0/adsl1
Selt results cleared on interface 1-4-3-0/adsl1
```

DELT (Dual-End Loop Test)

DELT is a dual-ended test that requires equipment that supports the DELT feature at both ends of the copper loop. While this prevents DELT from being used on loops where no CPE has yet been deployed, DELT offers a deeper set of loop tests, and can provide very valuable information on the condition of a copper loop. DELT is primarily used for reactive tests on a loop after a modem has been deployed to either help troubleshoot a line or capture a baseline of loop characteristics at the time of installation in order to assist in future troubleshooting. In addition, DELT can assist in predetermining if there is sufficient capability on that line to support new services, such as voice and video.

Starting DELT

DELT requires that there be an endpoint on the line to be tested. The endpoint equipment must also support DELT (e.g. Zhone CPE 6212). DELT is expected to be used in situations where the line quality may not be good enough for the port to train normally. Therefore, when performing DELT, the devices on each end of the line communicate more slowly than usual. It may take several minutes for the devices to exchange DELT information. Once the devices have shared DELT information, the line will return to an idle state.

Use the **delt start <interface>** command to start a DELT test on an ADSL2+ interface. To run DELT commands, the interface does not have to be down.

```
zSH> delt start 1-4-1-0/adsl1
Delt test started on interface 1-4-1-0/adsl1
```

Showing DELT status

Use the **delt show status <interface>** command to display DELT test progress on an ADSL2+ interface.

[Table 174](#) describes the DELT test status parameters.

Table 174: DELT test status parameters

Parameter	Description
Status	The status of the current DELT test. Values: in-progress The Loop Diagnostics process is in progress success The recent DELT command completed successfully aborted DELT was stopped, possible by the user none The DELT SNMP operation fails cannot-run Cannot start the test due to unspecified reason, possible the port is out of range illegal-mode The interface or line is in an illegal mode admin-up The interface has not been set administratively-down table-full DELT results table is full no-resource The system lacks a resource such as free memory
Device	The ADSL chip set in device used to perform the test.
Delt results generated <date>, <time>	The date and time of the test most-recently completed.
If the test is successful, the upstream and downstream values of the following parameters will be displayed:	
Attainable Bit Rate (bps)	Maximum Attainable Data Rate in bits per second. The maximum net data rate currently attainable by the ATU-C transmitter and ATU-R receiver (when referring to downstream direction) or by the ATU-R transmitter and ATU-C receiver (when referring to upstream direction).
Loop Attenuation (dB)	When referring to the downstream direction, it is the measured difference in the total power transmitted by the ATU-C and the total power received by the ATU-R over all sub-carriers during diagnostics mode. When referring to the upstream direction, it is the measured difference in the total power transmitted by the ATU-R and the total power received by the ATU-C over all sub-carriers during diagnostics mode. It ranges from 0 to 127 dB.
Signal Attenuation (dB)	When referring to the downstream direction, it is the measured difference in the total power transmitted by the ATU-C and the total power received by the ATU-R over all sub-carriers during SHOWTIME after the diagnostics mode. When referring to the upstream direction, it is the measured difference in the total power transmitted by the ATU-R and the total power received by the ATU-C over all sub carriers during SHOWTIME after the diagnostics mode. Range is 0 to 127 dB.

Table 174: DELT test status parameters (Continued)

Parameter	Description
SNR Margin (dB)	It is the maximum increase in dB of the noise power received at the ATU (ATU-R on downstream direction and ATU-C on upstream direction), such that bit error rate (BER) requirements are met for all bearer channels received at the ATU. It ranges from -64 to 63 dB.
Actual Transmit Power (dBm)	Actual Aggregate Transmit Power from the ATU (ATU-R on downstream direction and ATU-C on upstream direction), at the instant of measurement. It ranges from -31 to 31 dBm.

The following examples show the DELT status during the test and after the test completed:

```
zSH> delt show status 1-4-1-0/adsl1
Status: in-progress
Device: broadcom-6411
No delt results have been generated on this interface.
```

```
zSH> delt show status 1-4-1-0/adsl1
Status: success
Device: broadcom-6411
Delt results generated 30 jun 2009, 22:47:51.

                Downstream      Upstream
                -----
Attainable Bit Rate (bps)      23032000      1080000
Loop Attenuation (dB)          0.0          1.2
Signal Attenuation (dB)        0.0          0.2
SNR Margin (dB)                6.1          6.0
Actual Transmit Power (dBm)    16.1          11.3
```

Showing DELT noise

Use the **delt show noise <interface> [start-index [num-vals]]** command to display DELT noise, attenuation, and SNR per subcarrier.

The following describes the command options:

- <interface> can be in the form of ifIndex (e.g. 432), name/type (e.g. 1-4-1-0/adsl) or shelf/slot/port/subport/type (e.g. 1/4/1/0/adsl).
- [start-index: (0..511)] is the tone index with which to start.
- [num-vals]: the number of tones to display

[Table 175](#) describes the parameters in the DELT show noise command result.

Table 175: DELT show noise parameters

Parameter	Description
Tone Index	Tone index. In the range of 0 to 511. 0 indicates 4.3125 kHz Tone Freq, 1 indicates 8.625 kHz, ..., 511 indicates 2208.0000 kHz.
Tone Freq (kHz)	Tone frequency. Tone Freq = (Tone Index +1) x 4.3125 kHz.

Table 175: DELT show noise parameters (Continued)

Parameter	Description
If the test is successful, the upstream and downstream values of the following parameters will be displayed:	
Attenuation (dB)	An listing of up to 512 real H(f) logarithmic representation values in dB for the respective transmission direction. It supports up to 512 (downstream) sub-carriers. The number of utilized values for a direction (downstream or upstream) depends on the number of sub-carriers defined for that direction (NSCds or NSCus). Each row in the table contains a pair of Hlog(f = i*Df) values for a particular sub-carrier index. The real Hlog(f) value has a range of -96.3 to 6.0 dB. "no data" indicates that no measurement could be done for the sub-carrier because it is out of the passband or that the attenuation is out of range to be represented.
Noise (dBm/Hz)	An listing of up to 512 real Quiet Line Noise values in dBm/Hz for the respective transmission direction. Like Attenuation, it supports up to 512 (downstream) sub-carriers. Each row in the table contains a pair of QLN(f = i*Df) value values for a particular sub-carrier index. The QLN(f) value has a range of -150.0 to -23.0 dBm/Hz. "no data" indicates that no measurement could be done for the sub-carrier because it is out of the passband or that the noise PSD is out of range to be represented.
Signal-to-noise ratio (dB)	This is the SNR Margin per sub-carrier, expressing the ratio between the received signal power and received noise power per subscriber. It is a listing of 512 values. Like Attenuation, it supports up to 512 (downstream) sub-carriers. The SNR value has a range of -32 to 95 dB. "no data" indicates that no measurement could be done for the sub-carrier, probably because it is out of the PSD mask passband or that the noise PSD is out of range to be represented.

```
zSH> delt show noise 1-4-2-0/adsl
```

```
Delt results generated 30 jun 2009, 22:47:51.
```

Tone Index	Tone Freq (kHz)	Attenuation (dB)		Noise (dBm/Hz)		SNR (dB)	
		dnstream	upstream	dnstream	upstream	dnstream	upstream
0	4.3125	-78.1	-25.8	-84.0	-146.0	0.0	no data
1	8.6250	-40.7	-68.6	-127.0	-130.0	0.0	no data
2	12.9375	-42.9	-71.7	-127.0	-130.5	0.0	no data
3	17.2500	-42.9	-71.7	-127.0	-130.5	0.0	no data
4	21.5625	-45.3	-71.7	-127.0	-129.5	0.0	no data
5	25.8750	-48.7	-66.9	-127.0	-128.5	0.0	no data
6	30.1875	-48.7	-20.7	-127.0	-126.0	0.0	no data
7	34.5000	-42.6	-10.0	-127.0	-120.0	0.0	29.5
8	38.8125	-45.3	-0.6	-127.0	-113.5	0.0	35.5
9	43.1250	-51.2	1.5	-127.0	-113.0	0.0	41.0
10	47.4375	-51.7	1.7	-127.0	-112.0	0.0	43.5
11	51.7500	-54.3	2.3	-127.0	-113.5	0.0	46.0
12	56.0625	-51.2	2.9	-127.0	-114.5	0.0	48.0
13	60.3750	-54.3	3.2	-127.0	-111.5	0.0	49.0
14	64.6875	-47.6	3.2	-127.0	-112.5	0.0	51.0

```

15      69.0000      -47.7      2.9      -127.0      -110.5      0.0      51.5
[etc, up to index 511]

```

```
zSH> delt show noise 1-4-1-0/adsl 7 6
```

Delt results generated 30 jun 2009, 22:47:51.

Tone Index	Tone Freq (kHz)	Attenuation (dB)		Noise (dBm/Hz)		SNR (dB)	
		dnstream	upstream	dnstream	upstream	dnstream	upstream
7	34.5000	-42.6	-10.0	-127.0	-120.0	0.0	29.5
8	38.8125	-45.3	-0.6	-127.0	-113.5	0.0	35.5
9	43.1250	-51.2	1.5	-127.0	-113.0	0.0	41.0
10	47.4375	-51.7	1.7	-127.0	-112.0	0.0	43.5
11	51.7500	-54.3	2.3	-127.0	-113.5	0.0	46.0
12	56.0625	-51.2	2.9	-127.0	-114.5	0.0	48.0
13	60.3750	-54.3	3.2	-127.0	-111.5	0.0	49.0

Aborting DELT

Use the **delt abort <interface>** command to terminate a DELT test on an ADSL2+ interface.

Abort this test.

```
zSH> delt abort 1-4-1-0/adsl
```

Delt test aborted on interface 1-4-1-0/adsl

Clearing stored DELT results

Use the **delt clear <interface>** command to clear the stored DELT results on an ADSL2+ interface.

Clear the last DELT test results on the ADSL2+ interface.

```
zSH> delt clear 1-4-1-0/adsl
```

Delt results cleared on interface 1-4-1-0/adsl

14

MXK POTS CARDS

This chapter describes the MXK P-Phone POTS 24 card, POTS 72-port card, ADSL POTS combo card, and VDSL2+ POTS combo card:

- [P-phone POTS 24 card \(MXK-POTS-EBS-PKT-24\), page 1346](#)
- [POTS 72 card \(MXK-POTS-72\), page 1348](#)
- [POTS card configuration, page 1350](#)
- [ADSL+POTS combo cards \(MXK-ADSL2+-POTS-BCM-48A-2S, MXK-ADSL2+-POTS-BCM-48A-RNG-2S\), page 1361](#)
- [ADSL+POTS combo card configuration, page 1363](#)
- [VDSL2+POTS combo card \(MXK-VDSL2-POTS-BCM-17A-24\), page 1366](#)
- [VDSL+POTS combo card configuration, page 1367](#)
- [POTS interface configuration, page 1370](#)
- [Internal line testing and ring usage, page 1375](#)
- [POTS 24-port cards pinouts, page 1376](#)
- [POTS 72-port cards cable and port pinouts, page 1378](#)

For the voice configuration on the POTS and POTS combo cards, refer to Chapter [Voice Configuration, page 467](#).

P-phone POTS 24 card (MXK-POTS-EBS-PKT-24)



The MXK supports the 24 ports P-phone card (i.e. POTS-EBS card).

MXK-POTS-EBS-PKT-24 is a single slot card that supports POTS or EBS services within the SLMS system. Electronic Business Set (EBS) is a standard created by Nortel and is also known as P-Phone. The MXK-POTS-EBS-PKT-24 card supports packetized voice service for the POTS and EBS end-users when the MXK is subtended to a MALC with a voice gateway card.

This card supports the SIP-PLAR protocols.

Table 176 provides the specifications for the MXK-POTS-EBS-PKT-24.

Table 176: MXK POTS-EBS cards specifications

Specification	Value
Size	1 slot
Density	24 ports
Physical interfaces	One (1) RJ-21X Champ 50-pin connector
Redundancy	None
Nominal line rate	80 kbps 5 ppm

Table 176: MXK POTS-EBS cards specifications (Continued)

Specification	Value
Longitudinal balance:	500 Hz to 40 kHz: greater than 55 dB 40 kHz to 1 MHz: roll-off -20 dB per decade
Input return loss	greater than 20 dB, 10 kHz to 25 kHz roll-off 20 dB per decade to 1 kHz and 250 kHz
Free-run line rate (Stratum 4) if timing reference is lost	80 kbps 32 ppm
Metallic test functions	Look-out tests
Ring generation	Ring voltage supplied through Ring Voltage bus External generation through the External Ring Generator Input Port on TAC card.
Power consumption	31 w Nominal and 62 w Maximum

POTS 72 card (MXK-POTS-72)



MXK-POTS-72 card is a single slot card that supports packetized voice for use in a VoIP network. This card supports loop start, ground start, dial pulse, and provides echo cancellation. It has integrated ring generator as well as the internal line testing functionality (same capabilities as the TAC card) on the card.



Note: POTS 72 card cannot use TAC card to perform internal look-out line testing or generate ring. It uses the integrated testing functionality and ring generator instead.



Note: Cannot use external test head to perform line testing on the POTS 72 card.

MXK-POTS-72 card communicates with the uplink card over the MXK packet bus and the control bus.

This card supports the SIP, SIP-PLAR, MGCP, and MEGACO (H.248) protocols.

[Table 177](#) provides the specifications for the MXK-POTS-72.

Table 177: MXK POTS 72 cards specifications

Specification	Value
Size	1 slot
Density	72 ports
Physical interfaces	Two HDD78 78-pin connectors
Redundancy	None
Ring generation	Configurable Local Ringer Default balanced 85 VRMS w/23VDC offset
Metallic test functions	Integrated in the card, no TAC card needed
Nominal line rate	N/A
Longitudinal balance:	>53dB
Input return loss	>23dB
Voltage online (Tip to Ring)	49.5 V DC online
Current	25 MA
Loop Rating	1530 Ohms of loop
Power consumption	All lines on-hook 33W Add 1.3W per line off-hook 126W worst case power consumption.
POTS AC impedance	both 900 and 600 Ohms software settable

POTS card configuration

This section describes how to configure MXK POTS cards for IP packet voice. It includes:

- [Configuring 24-port POTS EBS cards on page 1350](#)
- [Configure a 72-port POTS card on page 1358](#)
- [Verifying the slot card installation on page 1360](#)

Each card installed in the system must have a **card-profile**. Each type of slot card requires different settings in the **card-profile**.



Tip: You can specify the name of the software image for a card in a **card-profile** or a **type-module**. Each card of a particular type can share a single **type-module**.

Settings in **type-modules** can be overridden by settings in **card-profiles**.

Each card in the system must have a **card-profile**. The line card type determines the parameter settings in the **card-profile** and the software image for the card. Performing a **card add** <slot #> automatically creates the **card-profile** for the card with the correct software image and settings.

The 24-port and 72-port POTS line cards on the MXK have the following card types and software images:

Table 178: MXK card types

Card	Type	Name of software image
MXK-POTS-EBS-PKT-24	10230	mxlc24ulcs.bin
MXK-POTS-72	10209	mxlc72pots.bin

Configuring 24-port POTS EBS cards

This section describes how to configure MXK-POTS-EBS-PKT-24 card for packet voice.

The following table describes the parameters in the **card-profile** for the MXK-POTS-EBS-PKT-24 card:

Parameter	Description
sw-file-name	Software image for the MXK-POTS-EBS-PKT-24 cards Values: mxlc24ulcs.bin
card-line-type	The type of calls supported on this card. Values: ebs-pots-pv EBS+POTS over packet voice.

Configuring a POTS-EBS card for packet voice

POTS-EBS card supports end users use the POTS phone or EBS phones (P-phones) and attendant console (key sets). The POTS-EBS card can provide packet voice through GR 303.

Figure 165: POTS-EBS card supports packet voice



Configure a POTS-EBS card that carrying the packet voice for both POTS and EBS, performing the following tasks:

- [Configuring the MALC with voice gateway card, page 1351](#)
- [Configuring the remote MXK with POTS-EBS card for both EBS and POTS end-users, page 1355](#)
- [Creating POTS and EBS voice connections in the class 5 switch, page 1358](#)
- [Viewing the detail information for the voice status in the remote MXK, page 1358](#)

Configuring the MALC with voice gateway card

To configure the MALC with voice gateway card:

- 1 Connect the physical T1 lines between the voicegateway card ports and the class 5 switch. This example uses a MALC-VG-T1/E1-32-2S card.

- 2 Create the voicegateway card card-profile with the *card-line-type* parameter of **ds1**. This reboots the voicegateway card.
This example has the VG card in slot 7.
- 3 Make sure all T1s are admin up on VG as by default they are down.
- 4 Update the system-clock-profile on the first two T1 lines on the VG card.

Use the **update system-clock-profile** command to update the *system-clock-eligibility* to **true** on the first two T1s.

```
zsh> update system-clock-profile 1-7-1-0/ds1
system-clock-profile 1-7-1-0/ds1
Please provide the following: [q]uit.
system-clock-eligibility: -> {false}: true
system-clock-weight: -----> {5}:
```

```
zsh> update system-clock-profile 1-7-2-0/ds1
system-clock-profile 1-7-2-0/ds1
Please provide the following: [q]uit.
system-clock-eligibility: -> {false}: true
system-clock-weight: -----> {5}:
```

- 5 Check **clkmgrshow**.

```
zsh> clkmgrshow
Primary system clock is 1/7/1/0 : T1
Secondary system clock is 1/7/2/0: T1
```

- 6 Note down the line group IDs, those line group IDs will be used for *dsn-lg-id* while creating GR303 interface group.

```
zsh> linegroup 1-7-1-0/ds1
linegroupId: 40
```

```
zsh> linegroup 1-7-2-0/ds1
linegroupId: 42
```

- 7 Use the **new gr303-interface-group** command to create a GR-303 interface group on the VG card.

```
zsh> new gr303-interface-group 1
gr303-interface-group 1
Please provide the following: [q]uit.
name-id: -----> {}: zhone
switch-type: -----> {}: norteldms100
adminStatus: -----> {}: inservice
working-mode: -----> {}: passive
ctrlChannel:
    control-channel-t303: -----> {700}
    control-channel-t396: -----> {14700}
    sapi-0-max-outstanding-frames: -> {7}
    sapi-0-n-200: -----> {3}
    sapi-0-t-200: -----> {150}
    sapi-0-t-203: -----> {30}
```

```

sapi-0-pps-mode: -----> {notinhibited}
sapi-1-max-outstanding-frames: -> {7}
sapi-1-n-200: -----> {3}
sapi-1-t-200: -----> {150}
sapi-1-t-203: -----> {30}
sapi-1-pps-mode: -> {notinhibited}
ds1LM has 32 elements. Display [a]ll, [n]one, a [s]ubset, or [q]uit? a
ds1LM[1]:
    dsn-lg-id: -----> {} 40 From line group id's recorded previously
    channel-number: -> {1}
    role: -----> {primary}
    logical-id: -----> {1}
    ds1-valid-flag: -> {valid}
ds1LM[2]:
    dsn-lg-id: -----> {} 42
    channel-number: -> {1}
    role: -----> {secondary}
    logical-id: -----> {2}
    ds1-valid-flag: -> {valid}
...

```

Refer to [Configuring a GR-303 interface on page 553](#) for the overall configuration procedure.

- 8** Create the entry of the profile voip-server-entry 255/255 with *zhoneVoipServerAddr* as **0.0.0.0** for the SIP PLAR voice connections.

```

zSH> new voip-server-entry 255/255
Please provide the following: [q]uit.
zhoneVoipServerAddrType: -----> {ipv4}:
zhoneVoipServerAddr: -----> {}: 0.0.0.0
zhoneVoipServerUdpPortNumber: -----> {5060}:
zhoneVoipServerId: -----> {generic}:
protocol: -----> {sip}:
sendCallProceedingTone: -----> {false}:
rtcpEnabled: -----> {false}:
rtcpPacketInterval: -----> {5000}:
interdigitTimeout: -----> {10}:
ipTos: -----> {0}:
systemDomainName: -----> {}:
expires-invite-value: -----> {3600}:
expires-register-value: -----> {3600}:
expires-header-method: -----> {register}:
session-timer: -----> {off}:
session-expiration: -----> {180}:
session-min-session-expiration: ---> {180}:
session-caller-request-timer: -----> {no}:
session-callee-request-timer: -----> {no}:
session-caller-specify-refresher: -> {omit}:
session-callee-specify-refresher: -> {uac}:
dtmf-mode: -----> {rfc2833}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```

- 9 Use the **new atm-traf-descr** command to create a new ATM traffic descriptor with an unique index for a voice connection.

```
zSH> new atm-traf-descr 1
atm-traf-descr 1
td_type: -----> {atmNoClpNoScr}
td_param1: -----> {0} 60000
td_param2: -----> {0}
td_param3: -----> {0}
td_param4: -----> {0}
td_param5: -----> {0}
cac-divider: -----> {1}
td_service_category: -----> {ubr}
usage-parameter-control: -> {true}
.....
Save new record? [s]ave, [c]hange or [q]uit: s
```

- 10 Use **new ip-interface-record vg/ip** and **new ip-unnumbered-record** commands to create an unnumbered interface for VoIP.

```
zSH> new ip-interface-record ebs/ip
vpi: -----> {0}:
vci: -----> {0}:
rdindex: -----> {1}:
dhcp: -----> {0.0.0.0} ** read-only**
addr: -----> {0.0.0.0} 10.235.9.1 Specify the
floater IP address for the VG
netmask: -----> {0.0.0.0} 255.255.255.0
bcastaddr: -----> {0.0.0.0} 10.235.9.255
destaddr: -----> {0.0.0.0}:
farendaddr: -----> {0.0.0.0}:
mru: -----> {1500}:
reasmmmaxsize: -----> {0}:
ingressfiltername: -> {}:
egressfiltername: --> {}:
pointtopoint: -----> {no}:
mcastenabled: -----> {yes}:
ipfwdenabled: -----> {yes}:
mcastfwdenabled: ---> {yes}:
natenabled: -----> {no}:
bcastenabled: -----> {yes}:
ingressfilterid: ---> {0}:
egressfilterid: ---> {0}:
ipaddrdynamic: -----> {static}:
dhcpserverenable: --> {false}:
subnetgroup: -----> {0}:
unnumberedindex: ---> {0}:
mcastcontrollist: --> {}:
vlanid: -----> {0}:
maxVideoStreams: ---> {0}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
```

```
zSH> new ip-unnumbered-record 1
```

```
ipUnnumberedInterfaceName: -> { }: ebs/ip
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

- 11** Use the **voicegateway add** command to create the voicegateway host using the available physical interface or slot number of the voicegateway card and traffic descriptor index.

```
zSH> voicegateway add -v 7 td 1 10.235.9.2
```

- 12** Check voice gateways.

```
zSH> vg show 7
```

Rd/Address	Interface	Group	T	Host Address
1 10.235.9.1	1-7-1-0-aal5proxy-0-32	0/32	0	S 10.235.9.2

- 13** Use the **voice add** command to add the VoIP to GR-303 voice connection between the voice gateway card and the switch.



Note: POTS or EBS is configured on a per line basis.

- a** For EBS voice:

```
zSH> voice add voip voip-1-7/ip dn 7311801 name 7311801 plar
172.24.200.52 reg 0 gr303 2/1801 ebs sub 7311801
```

- b** For POTS packet voice.

```
??zSH> voice add voip voip-1-7/ip dn 7311001 name 7311001 plar
172.24.200.51 reg 0 gr303 2/1001 sub 7311001
```

- 14** Use the **voice delete** command to delete the VoIP to GR-303 voice connection between the voice gateway card and the switch.

- a** If you want to delete an EBS to GR303 connection on the VG MALC, enter the following command.

```
zSH> voice delete voip voip-1-7/ip dn 7311801
```

- b** If you want to delete a POTS to GR303 connection on the VG MALC, enter the following command.

```
zSH> voice delete voip voip-1-7/ip dn 7311001
```

Configuring the remote MXK with POTS-EBS card for both EBS and POTS end-users

Configuration in the remote MXK with combo card, this combo card supports both EBS and POTS end-users with packet voice:

- 1** Create a uplink card **card-profile** for an uplink card on MXK.

2 View the type of card installed in the system:

```

zSH> slots
MXK 823
Uplinks
  a:*MXK FOUR GIGE (RUNNING+TRAFFIC)
  b: MXK FOUR GIGE (RUNNING)
Cards
  1: MXK 72 PORT POTS (RUNNING)
  2: MXK 72 PORT POTS (RUNNING)
  3: MXK 72 PORT POTS (RUNNING)
  4: MXK 72 PORT POTS (RUNNING)
  5: MXK 72 PORT POTS (RUNNING)
  6: MXK 72 PORT POTS (RUNNING)
 11: MXK 24 PORT ULCS/EBS POTS with Packet Voice
    (NOT_PROV)
 12: MXK 24 PORT ULCS/ISDN 2B1Q (RUNNING)
 14: MXK 24 PORT ULCS/ISDN 4B3T (RUNNING)
 15:*MTAC RING (RUNNING)

```

The POTS card in slot 11 is an MXK-POTS-EBS-PKT-24 card.

3 Create a **card-profile** for the MXK-POTS-EBS- PKT-24 card in *slot* 11, and specify the **card-line-type** to packet voice service (ebs-pots-pv):

```
zSH> card add 11 linetype ebs-pots-pv
```

or

```

zSH> new card-profile 1/11/10230
card-profile 1/11/10230
Please provide the following: [q]uit.
sw-file-name: -----> {mxlc24ulcs.bin}:
admin-status: -----> {operational}:
upgrade-sw-file-name: ----> {}:
upgrade-vers: -----> {}:
admin-status-enable: ----> {enable}:
sw-upgrade-admin: -----> {reloadcurrrev}:
sw-enable: -----> {true}:
sw-upgrade-enable: -----> {false}:
card-group-id: -----> {0}:
hold-active: -----> {false}:
weight: -----> {nopreference}:
card-line-type: -----> {unknowntype}: ebs-pots-pv
indicates plar packet voice
card-atm-configuration: -> {notapplicable}:
card-line-voltage: -----> {not-used}:
maxvpi-maxvci: -----> {notapplicable}:
card-init-string: -----> {}:
wetting-current: -----> {disabled}:
pwe-timing-mode: -----> {none}:
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.

```


- 4 In the remote MXK, use the **voice add** command to add the SIP-PLAR voice connection between the remote subtended MXK and the MALC with VG.
 - a To add an EBS to GR-303 connection on the remote MXK, make sure the **ulc-port-type** parameter in the **ulc-config profile** is ebs, and then use **voice add** command:

```
zSH> get ulc-config 1/11/1
```

```
ulc-port-type: ---> {ebs}
ulc-trap-enable: -> {disabled}
```

```
zSH> voice add ebs 1-11-1-0/voiceebs voip ethernet3-200/ip dn 7311801
name 7311801 plar 10.235.9.2 reg 0 sub 7311801 enable
Created subscriber-voice 1/7/44
Created subscriber-voice-ebs 167
Created subscriber-voice-voip 168
Interface 1-11-1-0/voiceebs's admin status is set to ENABLED
```

- b To add a POTS to GR-303 connection on the remote MXK, make sure **ulc-port-type** parameter in the **ulc-config profile** is pots, and then use **voice add** command:

```
zSH> update ulc-config 1/11/3
ulc-config 1/11/3
Please provide the following: [q]uit.
ulc-port-type: ---> {ebs}: pots
ulc-trap-enable: -> {disabled}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated
```

```
zSH> voice add pots 1-11-3-0/voicefxs voip ethernet3-200/ip dn 7311001
name 7311001 plar 10.235.9.2 reg 0 sub 7311001 enable
Created subscriber-voice 1/7/45
Created subscriber-voice-pots 169
Created subscriber-voice-voip 170
Interface 1-11-3-0's admin status is set to ENABLED
```

- 5 If you want to delete an EBS/POTS to GR303 connection on the remote MXK, enter the following command.
 - a To delete an EBS to GR-303 connection on the remote MXK, enter:

```
zSH> voice delete ebs 1-11-1-0/voiceebs
```

- b To delete a POTS to GR-303 connection on the remote MXK, enter:

```
zSH> voice delete pots 1-11-3-0/voicefxs
```

Creating POTS and EBS voice connections in the class 5 switch

Create EBS and POTS voice connections in the class 5 switch, for example, Nortel DMS 100. Refer to the Nortel user guide to get detail configuration.

Viewing the detail information for the voice status in the remote MXK

To view the detail information for the voice status (PLAR):

```
zSH> voice status 1 11 1
  port      term state  destination          call state  hook  ring  ESA
  ----      -
1-11-1-0/voiceebs  UP    VOIP:1634:VOIP EndPtIdx-168  No call    N/A NoRing  ON
Totals:  ports: 1, admin-state up: 1, active calls: 0

zSH> voice status 1 11 3
  port      term state  destination          call state  hook  ring  ESA
  ----      -
1-11-3-0/voicefxs  UP    VOIP:1634:VOIP EndPtIdx-170  No call    ON  NoRing  ON
Totals:  ports: 1, admin-state up: 1, active calls: 0
```

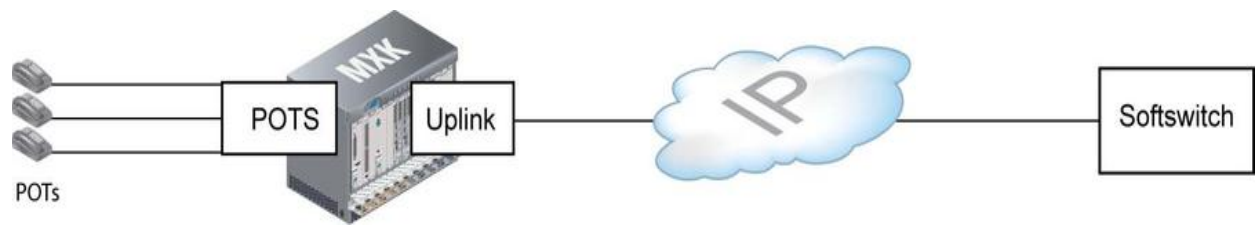
Configure a 72-port POTS card

This section describes how to configure MXK-POTS-72 card for packet voice.

The following table describes the parameters in the **card-profile** for the 72-port POTS card:

Parameter	Description
sw-file-name	Software image for the card. Values: mxlc72pots.bin
card-line-type	The type of calls supported on this card. Values: pots-pv POTS over packet voice.

Figure 166: POTS card supports packet voice



Configuring a 72-port POTS card for packet voice

The following example configure a MXK-POTS-72 card for packet voice:

- 1 View the type of card installed in the system:

```
zSH> slots
MXK 819
Uplinks
  b:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)

Cards
  3: MXK ADSL-48-A Bonded/with Packet Voice POTS
    (RUNNING)
  10:*MTAC RING (RUNNING)
  16: MXK 72 PORT POTS (NOT_PROV)
```

The POTS card in slot 16 is a MXK-POTS-72 card.

- 2 Create a **card-profile** for the MXK-POTS-72 card in *shelf 1, slot 16*, and specify the **card-line-type** to packet voice service:

```
zsh> card add 16 linetype pots-pv

or

zSH> new card-profile 16
card-profile 1/16/10209
Please provide the following: [q]uit.
sw-file-name: -----> {mxlc72pots.bin}:
admin-status: -----> {operational}:
upgrade-sw-file-name: ----> {}:
upgrade-vers: -----> {}:
admin-status-enable: ----> {enable}:
sw-upgrade-admin: -----> {reloadcurrrev}:
sw-enable: -----> {true}:
sw-upgrade-enable: -----> {false}:
card-group-id: -----> {0}:
hold-active: -----> {false}:
weight: -----> {nopreference}:
card-line-type: -----> {unknowntype}:pots-pv
indicates packet voice only
card-atm-configuration: -> {notapplicable}:
card-line-voltage: -----> {not-used}:
maxvpi-maxvci: -----> {notapplicable}:
card-init-string: -----> {}:
wetting-current: -----> {disabled}:
pwe-timing-mode: -----> {none}:
.....
Save changes? [s]ave, [c]hange or [q]uit:s
```

Verifying the slot card installation

After you save the **card-profile** record, the slot card in that slot resets and the begins downloading its software image from the uplink. This could take a few moments.

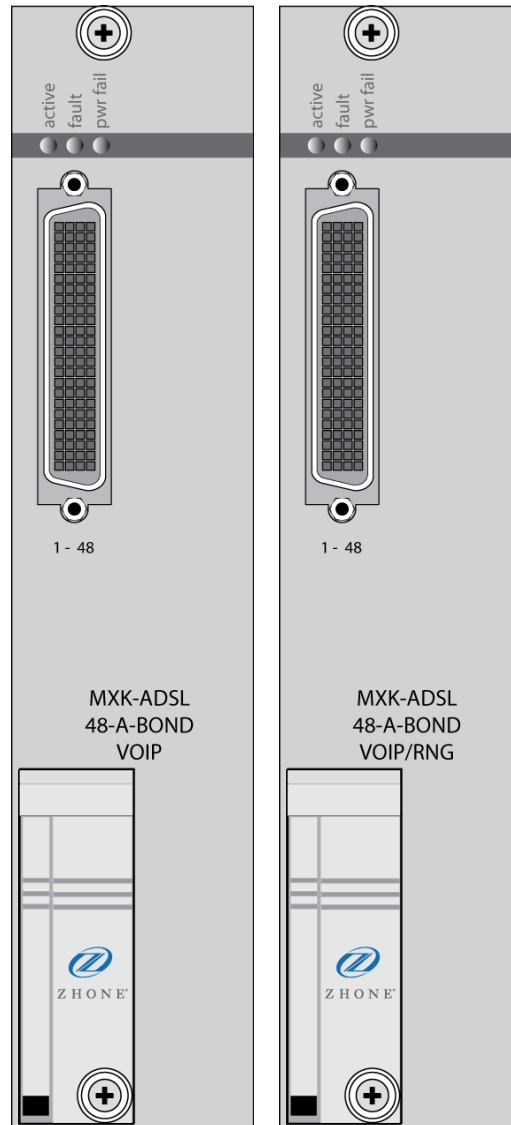
When the card has finished loading, a log message similar to the following is displayed (if logging is enabled):

```
zSH> Card in slot slot-number changed state to RUNNING
```

View card information including the state of the card and how long the card has been running.

```
zSH> slots 16
MXK 819
Type           : MXK 72 PORT POTS
Card Version   : 800-02810-01-Z
EEPROM Version : 16
Serial #       : 1262329
Card-Profile ID : 1/16/10209
Shelf          : 1
Slot           : 16
ROM Version    : MXK 2.1.227
Software Version: MXK 2.1.229
State          : RUNNING
Mode           : FUNCTIONAL
Heartbeat check : enabled
Heartbeat last  : WED SEP 29 16:00:06 2010
Heartbeat resp  : 80874
Heartbeat late  : 0
Hbeat seq error : 0
Hbeat longest   : 32
Fault reset     : enabled
  PF PF0        : 1
  PF PF1        : 1
  PF PF2        : 1
  PF PF3        : 1
  PF PF4        : 1
  PF PF5        : 1
  PF PF6        : 1
  PF PF7        : 1
# PF triggers   : 1
Uptime          : 4 minutes
Packet Voice    : Packet mode
```

ADSL+POTS combo cards (MXK-ADSL2+-POTS-BCM-48A-2S, MXK-ADSL2+-POTS-BCM-48A-RNG-2S)



The following MXK ADSL+ POTS combo cards can provide VoIP service:

- MXK-ADSL2+-POTS-BCM-48A-2S
- MXK-ADSL2+-POTS-BCM-48A-RNG-2S

These are two-slot cards that provide 48-ports of integrated ADSL and POTS VoIP services. These cards support the ANSI T1.413 Issue 2, G.992.1(G.dmt) and G.992.2 (G.lite), G.992.3 and G.992.4 (ADSL2), G.992.5 (ADSL2+), Annex A, and Annex M ADSL standards. Also supported are SIP, SIP-PLAR, MGCP, and H.248 (MEGACO) protocols.

Additionally, MXK-ADSL2+-POTS-BCM-48A-RNG-2S provides integrated ringing functionality and internal line testing functionality (same capabilities as the TAC ITM card).

Integrated ringing functionality on the line card means that the total number of POTS lines on an MXK chassis that can be in the ringing state simultaneously is increased as well as simplifying the effort required to match the ringing capacity of the TAC cards to the POTS lines on the shelf. Also, when the POTS lines in the chassis are provisioned on this card, a separate TAC card is not needed in the chassis, which increases the line capacity of the shelf and reduces the per port costs of deployment.



Note: MXK-ADSL2+-POTS-BCM-48A-RNG-2S card can use TAC card for test if necessary.



Note: MXK-ADSL2+-POTS-BCM-48A-RNG-2S card do not have access to the RING bus and therefore always use the integrated ring generator on the card.

Please note that the ADSL+ POTS combo card tech spec and its ATM services are described in [MXK ADSL2+ Bond Cards, page 1185](#).

ADSL+POTS combo card configuration

Each card installed in the system must have a **card-profile**. The line card type determines the parameter settings in the **card-profile** and the software image for the card. Performing a **card add <slot #>** automatically creates the **card-profile** for the card with the correct software image and settings.

Table 179 shows the card type and software image for the ADSL2+-POTS combo cards on the MXK.

Table 179: MXK ADSL POTS combo card types

Card	Type	Name of software image
MXK-ADSL2+-POTS-BCM-48A-2S	10202	<i>mxlc48aadslbond.bin</i>
MXK-ADSL2+-POTS-BCM-48A-RNG-2S	10202	<i>mxlc48badslbond.bin</i>

The card line types of 48-port ADSL2+ POTS combo cards on the MXK are:

- **unknowntype** (default)
- **adsl-pots-pv** (ADSL with VoIP)
- **adsl-pots-pv-rng-itm** (ADSL with VoIP, integrated ringing generation and line testing. For MXK-ADSL2+-POTS-BCM-48A-RNG-2S card only)

For MXK-ADSL2+-POTS-BCM-48A-RNG-2S card:

- Both **adsl-pots-pv** or **adsl-pots-pv-rng-itm** linetypes use the internal ring generator on the card.
- By provisioning a card line type parameter to **adsl-pots-pv** for the MXK-ADSL2+-POTS-BCM-48A-RNG-2S card, it will cause this RNG combo card to behave exactly as the non-RNG versions of ADSL POTS combo cars from a loop test perspective. In this mode therefore, loop testing can be achieved through external test heads (like Tollgrade) from test access ports on the TAC cards. Alternatively, you can use the integrated Test Module (ITM) functionality on the TAC cards to perform look out testing on the RNG combo cards.
- By provisioning a card line type parameter to **adsl-pots-pv-rng-itm** for the RNG combo card, it will cause the RNG combo card use the integrated ITM test functionality on the RNG combo card. In this mode, access to the test functionality of the TAC cards (either the ITM or the external test access ports) is blocked.

Adding ADSL-POTS-RNG-Combo cards

The following example adds an MXK-ADSL2+-POTS-BCM-48A-RNG-2S card to the system:

- 1 Install the MXK-ADSL2+-POTS-BCM-48A-RNG-2S card in the desired line card slot.

2 Create a **card-profile** for the card, and specify the linetype:

```
zSH> card add 10 linetype adsl-pots-pv-rng-itm
```

Or

```
zSH> new card-profile 10
card-profile 1/10/10202
sw-file-name: -----> {mxlc48aadslbond.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {unknowntype}adsl-pots-pv-rng-itm
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}
.....
Save new record? [s]ave, [c]hange or [q]uit: s
New record saved.
```

3 Verify the card by entering **slots**:

```
zSH> slots
MXK 819
Uplinks
  a:*MXK EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK EIGHT GIGE (NOT_PROV)
Cards
  1: MXK ADSL-48-A Bonded (RUNNING)
  2: MXK ADSL-48-A Bonded (RUNNING)
  3: MXK ADSL-48-A Bonded (RUNNING)
  4: MXK ADSL-48-A Bonded (RUNNING)
  5: MXK ADSL-48-A Bonded (RUNNING)
  6: MXK ADSL-48-A Bonded (RUNNING)
  7: MXK ADSL-48-A Bonded (RUNNING)
  8: MXK ADSL-48-A Bonded (RUNNING)
  9: MXK ADSL-48-A Bonded (RUNNING)
 10: MXK ADSL-48-A Bonded/with Packet Voice POTS, RNG, ITM (RUNNING)
 11: MXK ADSL-48-A Bonded/with 900 Ohm Splitter (RUNNING)
 13: MXK ADSL-48-A Bonded/with 900 Ohm Splitter (RUNNING)
 15: MXK ADSL-48-A Bonded/with 600 Ohm Splitter (RUNNING)
 17: MXK ADSL-48-A Bonded/with 600 Ohm Splitter (RUNNING)
```

4 Verify the **card-profile** for the ADSL card:


```

zSH> get card-profile 1/10/10202
card-profile 1/10/10202
sw-file-name: -----> {mx1c48aads1bond.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {adsl-pots-pv-rng-itm}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}

```

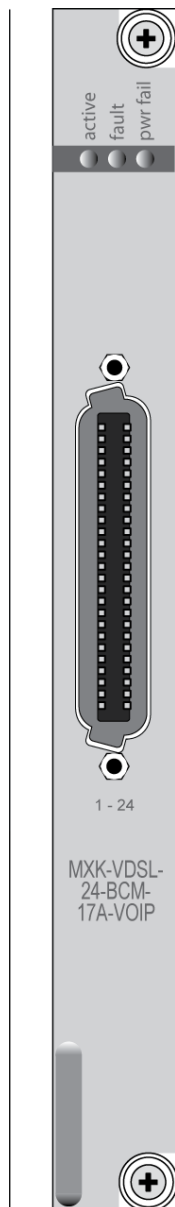
5 View card information including the state of the card and how long the card has been running:

```

zSH> slots 10
MXK 819
Type           : MXK ADSL-48-A Bonded
Sub-Type       : with Packet Voice POTS, RNG, ITM
Card Version   : 800-02968-01-B
EEPROM Version : 1
Serial #       : 4069337
CLEI Code      : No CLEI
Card-Profile ID : 1/10/10202
Shelf          : 1
Slot           : 10
ROM Version    : MXK 2.0.100
Software Version: MXK 2.1.208
State          : RUNNING
Mode           : FUNCTIONAL
Heartbeat check : enabled
Heartbeat last  : WED AUG 18 16:22:21 2010
Heartbeat resp  : 1229506
Heartbeat late  : 0
Hbeat seq error : 0
Hbeat longest   : 10
Fault reset     : enabled
Uptime          : 13 days, 8 hours, 25 minutes
Packet Voice    : Packet mode

```

VDSL2+POTS combo card (MXK-VDSL2-POTS-BCM-17A-24)



The MXK-VDSL2-POTS-BCM-17A-24 card provides 24 ports of integrated VDSL2 and POTS VoIP services. In addition to the standards listed in [Table 177](#), this card also supports SIP, SIP-PLAR, H.248, MGCP protocols and H.248 (MEGACO) protocols.

This chapter provides POTS VoIP service configuration for the VDSL POTS combo card.

VDSL+POTS combo card configuration

Each card installed in the system must have a **card-profile**. The line card type determines the parameter settings in the **card-profile** and the software image for the card. Performing a **card add** <slot #> automatically creates the **card-profile** for the card with the correct software image and settings.

Table 179 shows the card type and software image for the VDSL2 POTS combo cards on the MXK.

Table 180: MXK ADSL POTS combo card types

Card	Type	Name of software image
MXK-VDSL2-POTS-BCM-17A-24	10222	<i>mxlc24vdsl2pots.bin</i>

Adding VDSL POTS combo cards

The following example adds an MXK-VDSL2-POTS-BCM-17A-24 card to the system:

- 1 Install the MXK-VDSL2-POTS-BCM-17A-24 card in the desired line card slot.
- 2 Create a **card-profile** for the card:

```
zSH> card add 13
```

After performing a **card add** in a slot, the slot resets and begins downloading the software image from the flash card. This could take a few moments.

When the card has finished loading, a log message similar to the following is displayed (if logging is enabled):

```
zSH> Card in slot slot-number changed state to RUNNING
```

- 3 Verify the card by entering **slots**:

```
zSH> slots
MXK 819
Uplinks
  a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)
Cards
  1: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
  2: MXK 20 ACT ETH (RUNNING)
  4: MXK ADSL-48-A Bonded (RUNNING)
  5: MXK 4 PORT GPON (RUNNING)
  6: MXK 4 PORT GPON (RUNNING)
  7: MXK 20 ACT ETH SINGLE SLOT (RUNNING)
  8: MXK ADSL-48-A Bonded/with Packet Voice POTS (RUNNING)
 10: MXK ADSL-48-A Bonded (RUNNING)
 13: MXK 24 PORT VDSL2 POTS (RUNNING)
 14:*TAC (RUNNING)
```

4 Verify the **card-profile** for the VDSL POTS combo card:

```

zSH> get card-profile 1/13/10222
card-profile 1/13/10222
sw-file-name: -----> {mxlc24vdsl2pots.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {unknownntype}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}

```

5 View card information including the state of the card and how long the card has been running:

```

zSH> slots 13
MXK 819
Type           : MXK 24 PORT VDSL2 POTS
Card Version    : 800-02975-02-A
EEPROM Version  : 1
Serial #        : 4059003
Card-Profile ID : 1/13/10222
Shelf           : 1
Slot            : 13
ROM Version     : development
Software Version: MXK 2.2.1.138
State           : RUNNING
Mode            : FUNCTIONAL
Heartbeat check : enabled
Heartbeat last  : MON SEP 18 12:05:36 2000
Heartbeat resp  : 1060341
Heartbeat late  : 0
Hbeat seq error : 0
Hbeat longest   : 33
Fault reset     : enabled
Power fault mon : supported
  PF +12V       : 1
  PF 1.8V       : 1
  PF 1.2V & 3.3V : 1
  PF 2.5V       : 1
  PF 1.5V       : 1
  PF PF5        : 1
  PF PF6        : 1

```

PF daughter : 1
PF triggers : 1
Uptime : 35 minutes
Packet Voice : Packet mode

POTS interface configuration

This section describes how to configure POTS interfaces. It includes:

- [Configuring POTS settings on page 1370](#)
- [Configuring signal type and ring frequency on page 1373](#)

The following table summarizes how to configure POTS interfaces on the MXK:

Action	Command
Configure the POTS gain settings. See Configuring POTS settings on page 1370 .	update analog-if-cfg-profile <i>index</i>/voicefxs Where <i>index</i> is of the form <i>shelf-slot-port-subport</i> or a user-defined string. For typical applications, the settings in this profile do not need to be modified.
Configure the POTS signaling. See Configuring signal type and ring frequency on page 1373 .	update analog-fxs-cfg-profile <i>index</i>/voicefxs For typical applications, the settings in this profile do not need to be modified.

Configuring POTS settings

Modify the following parameters in the **analog-if-cfg-profile** if you need to change the gain settings for each voice line:

Parameter	Description
if-cfg-impedence	Specifies the terminating impedance of analog voice interfaces. Values: ohms600complex 600 Ohms + 2.16uF ohms900complex 900 Ohms + 2.16uF Default: ohms600complex
if-cfg-receive-tp	The receive TLP is the signal level to the customer premises equipment (CPE). The receive signal range is +3 dB to -9 dB. A positive number adds gain, a negative number adds loss to the analog signal after decoding from PCM. For example, a receive TLP setting of -6 dB will generate a voice signal at -6 dB level. Values: fxsrtlpn9db fxsrtlpn8db fxsrtlpn7db fxsrtlpn6db fxsrtlpn5db fxsrtlpn4db fxsrtlpn3db fxsrtlpn2db fxsrtlpn1db fxsrtlp0db fxsrtlp1db fxsrtlp2db fxsrtlp3db rtlpnumeric Default: fxsrtlpn6db

Parameter	Description
if-cfg-transmit-ttp	The transmit TLP is the signal level from the customer premises equipment (CPE). The transmit signal range is +9 dB to -3 dB. A positive number adds loss, a negative number adds gain to the analog signal before encoding to PCM. For example, a transmit TLP setting of +3 dB will set a loss of 3 dB to generate a 0 dB PCM signal. Values: fxsTtlp9db fxsTtlp8db fxsTtlp7db fxsTtlp6db fxsTtlp5db fxsTtlp4db fxsTtlp3db fxsTtlp2db fxsTtlp1db fxsTtlp0db fxsTtlpN1db fxsTtlpN2db fxsTtlpN3db Default: fxsTtlp0db
if-cfg-pcm-encoding	Line encoding. Select one match to the country setting. Values: alaw for E1. mulaw for T1. Default: mulaw
if-cfg-receive-ttpNum	Receive Transmission Level Point (RTLTP) settings control the amount gain or loss added to the incoming signal after it is decoded to analog. To increase the signal level set the RTLTP setting to higher values. The default is 0 dB. Values: -160 to 85 (in tenths of dB) Default: 0
if-cfg-transmit-ttpNum	Transmit Transmission Level Point (TTLP) controls the amount of gain or loss added to a voice signal before it is encoded to digital PCM. To increase the signal level, reduce the TTLP setting to lower value. Values: -175 to 70 (in tenths of dB) Default: 0

Parameter	Description
if-cfg-loop-current	Adjust the loop current of an analog pots line. This parameter is applicable to the MXK combo POTS 48 cards (i.e. equipped with local ringing daughter card) and the MXK POTS 72 card. Values: 20 to 44 (in mA) Default: 30
if-cfg-ring-voltage	Adjust the ringing voltage of an analog pots line. This parameter is applicable to the MXK combo POTS 48 cards (i.e. equipped with local ringing daughter card) and the MXK POTS 72 card. Values: b85vrms b75vrms b92vrms Default: b85vrms

If you need to modify the gain settings, update the **analog-if-cfg-profile** for each interface. For example:

```

zSH> update analog-if-cfg-profile 1-3-1-0/voicefxs
Please provide the following: [q]uit)
if-cfg-impedence: ----->{ohms600complex}: modify if required
if-cfg-receive-tlp: ----->{fxsrtlpn6db}: modify if required
if-cfg-transmit-tlp: ----->{fxsttlp0db}: modify if required
if-cfg-trunk-conditioning: --->{idle}:
if-maintenance-mode: ----->{off}:
if-cfg-pcm-encoding: ----->{mulaw}: alaw
if-cfg-receive-tlpNum: ----->{0}:
if-cfg-transmit-tlpNum: ----->{0}:
if-cfg-loop-current: ----->{30}:
if-cfg-ring-voltage: ----->{b85vrms}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Configuring signal type and ring frequency

Modify the following parameters in the **analog-fxs-cfg-profile** if you need to change signalling type and ring frequency for each voice line:

Parameter	Description
signal-type	The method by which an off-hook condition is indicated. Values: fxsloopstart fxsgroundstart Default: fxsloopstart
ring-frequency	Rate in cycles per second (Hertz) at which polarity reversal occurs on ringing. Values: ringfrequency20 ringfrequency25 ringfrequency30 ringfrequency50 Default: ringfrequency20
ring-back	The ring back is requested if this variable is set to on. Values: on off Default: off

If you need to modify the signaling and ring frequency, update the **analog-fxs-cfg-profile** for each interface. For example:

```
zSH> update analog-fxs-cfg-profile 1-3-1-0/voicefxs
```

```
analog-fxs-cfg-profile 1-3-1-0/voicefxs
please provide the following: [q]uit.
signal-type: ----> {fxsloopstart} fxsgroundstart
ring-frequency: -> {ringfrequency20}
ring-back: -----> {off}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Internal line testing and ring usage

The following table shows the test and ring usage for MXK P-Phone POTS 24 card (i.e. 24-Port POTS & EBS card), POTS 72-port card, ADSL POTS combo card, and VDSL2+ POTS combo card.

Table 181: POTS cards supporting line test, ring generator, SELT/DELT

Card Grouping	Card Name	Internal Line Tests	Line Tests access to External Test Head Via TAC card	Ring using Integrated Ring Generator	SELT/DELT Support
48-Port ADSL2 & POTS	MXK-ADSL2+-POTS-BCM-48A-2S	Via MXK-TAC-IT M-RING card	Yes	Via MXK-TAC-ITM-RING card	Yes
	MXK-ADSL2+-POTS-BCM-48A-RNG-2S (With card line type = adsl-pots-pv)	Via MXK-TAC-IT M-RING card	Yes	Via On-Card ring generator	
	MXK-ADSL2+-POTS-BCM-48A-RNG-2S (With card line type = adsl-pots-pv-rng-itm)	Via On-Card Integrated Test Module (ITM)	No	Via On-Card ring generator	
72-Port POTS	MXK-POTS-72	Via On-Card ITM	No	Via On-Card ring generator	N/A
24-Port POTS & EBS	MXK-POTS-EBS-PKT-24	Via MXK-TAC-IT M-RING card	Yes	Via MXK-TAC-ITM-RING card	N/A
24-Port VDSL2 & POTS	MXK-VDSL2-POTS-BCM-17A-24	Via On-Card ITM	No	Via On-Card ring generator	Yes

Note that the MXK-POTS-72 card, MXK-VDSL2-POTS-BCM-17A card, and the MXK-ADSL2+-POTS-BCM-48A-RNG-2S card have integrated loop testing functionality. The integrated test functionality on the MXK-ADSL2+-POTS-BCM-48A-RNG-2S card can be turned on by selecting the card line type adsl-pots-pv-rng-itm.

The commands to perform internal line testing on the those card with the integrated line testing module are same as with the MXK-TAC-ITM-RING card.

For the commands and testing procedures, refer to [MXK Test Access Cards, page 1523](#).

POTS 24-port cards pinouts

The MXK-POTS-EBS-PKT-24 card use standard RJ-21X pinouts.

Figure 168 shows the location of pinouts on the POTS 24 card connector.

Figure 167: 24-port POTS card connector pinouts

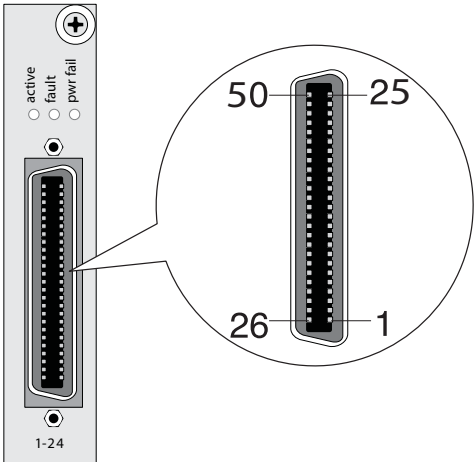


Table 182 lists the MXK-POTS-EBS-PKT-24 card pinouts.

Table 182: 24-port POTS card pinouts

Pin	Function	Pin	Function
1	Channel 1 ring	26	Channel 1 tip
2	Channel 2 ring	27	Channel 2 tip
3	Channel 3 ring	28	Channel 3 tip
4	Channel 4 ring	29	Channel 4 tip
5	Channel 5 ring	30	Channel 5 tip
6	Channel 6 ring	31	Channel 6 tip
7	Channel 7 ring	32	Channel 7 tip
8	Channel 8 ring	33	Channel 8 tip
9	Channel 9 ring	34	Channel 9 tip
10	Channel 10 ring	35	Channel 10 tip
11	Channel 11 ring	36	Channel 11 tip
12	Channel 12 ring	37	Channel 12 tip
13	Channel 13 ring	38	Channel 13 tip
14	Channel 14 ring	39	Channel 14 tip
15	Channel 15 ring	40	Channel 15 tip

Table 182: 24-port POTS card pinouts (Continued)

Pin	Function	Pin	Function
16	Channel 16 ring	41	Channel 16 tip
17	Channel 17 ring	42	Channel 17 tip
18	Channel 18 ring	43	Channel 18 tip
19	Channel 19 ring	44	Channel 19 tip
20	Channel 20 ring	45	Channel 20 tip
21	Channel 21 ring	46	Channel 21 tip
22	Channel 22 ring	47	Channel 22 tip
23	Channel 23 ring	48	Channel 23 tip
24	Channel 24 ring	49	Channel 24 tip
25	Not used	50	Not used

POTS 72-port cards cable and port pinouts

This section describes the 72-port POTS cards pinouts and the 72-port POTS cable available from Zhone Technologies.

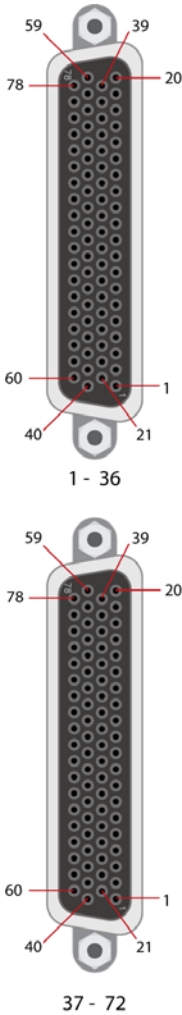
- [POTS 72-port card port pinouts, page 1378](#)
- [POTS 72-port card cable pinouts, page 1384](#)

POTS 72-port card port pinouts

The MXK-POTS-72 card uses two standard HDD78 connectors.

[Figure 168](#) shows the location of pinouts on the MXK-POTS-72 card connectors.

Figure 168: 72-port POTS card connector pinouts



[Table 183](#) lists the MXK-POTS-72 card pinouts of the top connector.

Table 183: 72-port POTS card pinouts - the top connector

Port	Signal	Pin
1	Tip	71
	Ring	70
2	Tip	73
	Ring	72
3	Tip	75
	Ring	74
4	Tip	77
	Ring	76
5	Tip	37
	Ring	38
6	Tip	35
	Ring	36
7	Tip	33
	Ring	34
8	Tip	31
	Ring	32
9	Tip	53
	Ring	52
10	Tip	55
	Ring	54
11	Tip	57
	Ring	56
12	Tip	59
	Ring	58
13	Tip	19
	Ring	20
14	Tip	17
	Ring	18
15	Tip	15
	Ring	16

Table 183: 72-port POTS card pinouts - the top connector

Port	Signal	Pin
16	Tip	13
	Ring	14
17	Tip	11
	Ring	12
18	Tip	9
	Ring	10
19	Tip	7
	Ring	8
20	Tip	29
	Ring	30
21	Tip	27
	Ring	28
22	Tip	25
	Ring	26
23	Tip	23
	Ring	24
24	Tip	21
	Ring	22
25	Tip	5
	Ring	6
26	Tip	3
	Ring	4
27	Tip	1
	Ring	2
28	Tip	41
	Ring	40
29	Tip	67
	Ring	66
30	Tip	65
	Ring	64

Table 183: 72-port POTS card pinouts - the top connector

Port	Signal	Pin
31	Tip	63
	Ring	62
32	Tip	61
	Ring	60
33	Tip	43
	Ring	42
34	Tip	45
	Ring	44
35	Tip	47
	Ring	46
36	Tip	49
	Ring	48

[Table 184](#) lists the MXK-POTS-72 card pinouts of the bottom connector.

Table 184: 72-port POTS card pinouts - the bottom connector

Port	Signal	Pin
37	Tip	71
	Ring	70
38	Tip	73
	Ring	72
39	Tip	75
	Ring	74
40	Tip	77
	Ring	76
41	Tip	37
	Ring	38
42	Tip	35
	Ring	36

Table 184: 72-port POTS card pinouts - the bottom connector

Port	Signal	Pin
43	Tip	33
	Ring	34
44	Tip	31
	Ring	32
45	Tip	53
	Ring	52
46	Tip	55
	Ring	54
47	Tip	57
	Ring	56
48	Tip	59
	Ring	58
49	Tip	13
	Ring	14
50	Tip	15
	Ring	16
51	Tip	17
	Ring	18
52	Tip	19
	Ring	20
53	Tip	21
	Ring	22
54	Tip	23
	Ring	24
55	Tip	25
	Ring	26
56	Tip	27
	Ring	28
57	Tip	11
	Ring	12

Table 184: 72-port POTS card pinouts - the bottom connector

Port	Signal	Pin
58	Tip	9
	Ring	10
59	Tip	7
	Ring	8
60	Tip	5
	Ring	6
61	Tip	3
	Ring	4
62	Tip	1
	Ring	2
63	Tip	29
	Ring	30
64	Tip	61
	Ring	60
65	Tip	63
	Ring	62
66	Tip	65
	Ring	64
67	Tip	67
	Ring	66
68	Tip	41
	Ring	40
69	Tip	43
	Ring	42
70	Tip	45
	Ring	44
71	Tip	47
	Ring	46
72	Tip	49
	Ring	48

POTS 72-port card cable pinouts

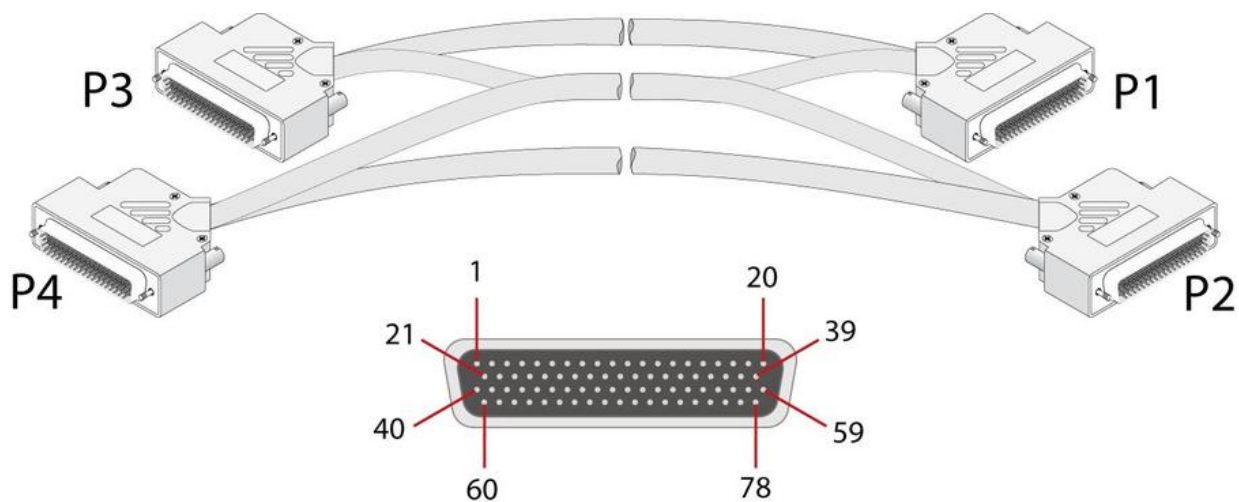
This section provides the following information:

- [Dual 78-pin to dual 78-pin connector cable, page 1384](#)
- [Dual 78-pin to three 50-pin connector cable, page 1391](#)
- [Dual 78-pin to blunt connector cable, page 1399](#)

Dual 78-pin to dual 78-pin connector cable

[Figure 169](#) shows the dual 78-pin to dual 78-pin connector cable used for 72-port POTS card. [Table 185 on page 1384](#) lists variations of these cables. [Table 186 on page 1386](#) lists pinouts of these cables.

Figure 169: dual 78-pin to dual 78-pin cable



Variations of dual 78-pin to dual 78-pin connector cables

[Table 185](#) describes the dual 78-pin to dual 78-pin connector cables.

Table 185: dual 78-pin to dual 78-pin connector cables

MXK CABLE PART NAME	DESCRIPTION
MXK-CBL-72-72-CAT3-4FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 4FT/1.22M
MXK-CBL-72-72-CAT3-10FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 10FT/3.05M
MXK-CBL-72-72-CAT3-15FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 15FT/4.57M
MXK-CBL-72-72-CAT3-30FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 30FT/9.10M

Table 185: dual 78-pin to dual 78-pin connector cables (Continued)

MXK CABLE PART NAME	DESCRIPTION
MXK-CBL-72-72-CAT3-50FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 50FT/15.23M
MXK-CBL-72-72-CAT3-100FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 100FT/30.47M
MXK-CBL-72-72-CAT5-4FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 4FT/1.22M
MXK-CBL-72-72-CAT5-10FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 10FT/3.05M
MXK-CBL-72-72-CAT5-15FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 15FT/4.57M
MXK-CBL-72-72-CAT5-30FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 30FT/9.10M
MXK-CBL-72-72-CAT5-50FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 50FT/15.23M
MXK-CBL-72-72-CAT5-100FT	CABLE: (2) 78PIN TO (2) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 100FT/30.47M

dual 78-pin to dual 78-pin cable pinouts

[Table 186](#) lists pinouts of the dual 78-pin to dual 78-pin cables.

Table 186: dual 78-pin to dual 78-pin cable pinouts

Pair	Signal	Color	From	To	Cable
1	Tip	White/Blue	P1-71	P3-71	Cable 1
	Ring	Blue/White	P1-70	P3-70	
2	Tip	White/Orange	P1-73	P3-73	
	Ring	Orange/White	P1-72	P3-72	
3	Tip	White/Green	P1-75	P3-75	
	Ring	Green/White	P1-74	P3-74	
4	Tip	White/Brown	P1-77	P3-77	
	Ring	Brown/White	P1-76	P3-76	
5	Tip	White/Slate	P1-37	P3-37	
	Ring	Slate/White	P1-38	P3-38	
6	Tip	Red/Blue	P1-35	P3-35	
	Ring	Blue/Red	P1-36	P3-36	
7	Tip	Red/Orange	P1-33	P3-33	
	Ring	Orange/Red	P1-34	P3-34	
8	Tip	Red/Green	P1-31	P3-31	
	Ring	Green/Red	P1-32	P3-32	
9	Tip	Red/Brown	P1-53	P3-53	
	Ring	Brown/Red	P1-52	P3-52	
10	Tip	Red/Slate	P1-55	P3-55	
	Ring	Slate/Red	P1-54	P3-54	
11	Tip	Black/Blue	P1-57	P3-57	
	Ring	Blue/Black	P1-56	P3-56	
12	Tip	Black/Orange	P1-59	P3-59	
	Ring	Orange/Black	P1-58	P3-58	
13	Tip	Black/Green	P1-19	P3-19	
	Ring	Green/Black	P1-20	P3-20	
14	Tip	Black/Brown	P1-17	P3-17	
	Ring	Brown/Black	P1-18	P3-18	

Table 186: dual 78-pin to dual 78-pin cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
15	Tip	Black/Slate	P1-15	P3-15	Cable 1 (Continued)
	Ring	Slate/Black	P1-16	P3-16	
16	Tip	Yellow/Blue	P1-13	P3-13	
	Ring	Blue/Yellow	P1-14	P3-14	
17	Tip	Yellow/Orange	P1-11	P3-11	
	Ring	Orange/Yellow	P1-12	P3-12	
18	Tip	Yellow/Green	P1-9	P3-9	
	Ring	Green/Yellow	P1-10	P3-10	
19	Tip	Yellow/Brown	P1-7	P3-7	
	Ring	Brown/Yellow	P1-8	P3-8	
20	Tip	Yellow/Slate	P1-29	P3-29	
	Ring	Slate/Yellow	P1-30	P3-30	
21	Tip	Violet/Blue	P1-27	P3-27	
	Ring	Blue/Violet	P1-28	P3-28	
22	Tip	Violet/Orange	P1-25	P3-25	
	Ring	Orange/Violet	P1-26	P3-26	
23	Tip	Violet/Green	P1-23	P3-23	
	Ring	Green/Violet	P1-24	P3-24	
24	Tip	Violet/Brown	P1-21	P3-21	
	Ring	Brown/Violet	P1-22	P3-22	

Table 186: dual 78-pin to dual 78-pin cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
25	Tip	White/Blue	P1-5	P3-5	Cable 2
	Ring	Blue/White	P1-6	P3-6	
26	Tip	White/Orange	P1-3	P3-3	
	Ring	Orange/White	P1-4	P3-4	
27	Tip	White/Green	P1-1	P3-1	
	Ring	Green/White	P1-2	P3-2	
28	Tip	White/Brown	P1-41	P3-41	
	Ring	Brown/White	P1-40	P3-40	
29	Tip	White/Slate	P1-67	P3-67	
	Ring	Slate/White	P1-66	P3-66	
30	Tip	Red/Blue	P1-65	P3-65	
	Ring	Blue/Red	P1-64	P3-64	
31	Tip	Red/Orange	P1-63	P3-63	
	Ring	Orange/Red	P1-62	P3-62	
32	Tip	Red/Green	P1-61	P3-61	
	Ring	Green/Red	P1-60	P3-60	
33	Tip	Red/Brown	P1-43	P3-43	
	Ring	Brown/Red	P1-42	P3-42	
34	Tip	Red/Slate	P1-45	P3-45	
	Ring	Slate/Red	P1-44	P3-44	
35	Tip	Black/Blue	P1-47	P3-47	
	Ring	Blue/Black	P1-46	P3-46	
36	Tip	Black/Orange	P1-49	P3-49	
	Ring	Orange/Black	P1-48	P3-48	

Table 186: dual 78-pin to dual 78-pin cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
37	Tip	Black/Green	P2-71	P4-71	Cable 2 (Continued)
	Ring	Green/Black	P2-70	P4-70	
38	Tip	Black/Brown	P2-73	P4-73	
	Ring	Brown/Black	P2-72	P4-72	
39	Tip	Black/Slate	P2-75	P4-75	
	Ring	Slate/Black	P2-74	P4-74	
40	Tip	Yellow/Blue	P2-77	P4-77	
	Ring	Blue/Yellow	P2-76	P4-76	
41	Tip	Yellow/Orange	P2-37	P4-37	
	Ring	Orange/Yellow	P2-38	P4-38	
42	Tip	Yellow/Green	P2-35	P4-35	
	Ring	Green/Yellow	P2-36	P4-36	
43	Tip	Yellow/Brown	P2-33	P4-33	
	Ring	Brown/Yellow	P2-34	P4-34	
44	Tip	Yellow/Slate	P2-31	P4-31	
	Ring	Slate/Yellow	P2-32	P4-32	
45	Tip	Violet/Blue	P2-53	P4-53	
	Ring	Blue/Violet	P2-52	P4-52	
46	Tip	Violet/Orange	P2-55	P4-55	
	Ring	Orange/Violet	P2-54	P4-54	
47	Tip	Violet/Green	P2-57	P4-57	
	Ring	Green/Violet	P2-56	P4-56	
48	Tip	Violet/Brown	P2-59	P4-59	
	Ring	Brown/Violet	P2-58	P4-58	

Table 186: dual 78-pin to dual 78-pin cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
49	Tip	White/Blue	P2-13	P4-13	Cable 3
	Ring	Blue/White	P2-14	P4-14	
50	Tip	White/Orange	P2-15	P4-15	
	Ring	Orange/White	P2-16	P4-16	
51	Tip	White/Green	P2-17	P4-17	
	Ring	Green/White	P2-18	P4-18	
52	Tip	White/Brown	P2-19	P4-19	
	Ring	Brown/White	P2-20	P4-20	
53	Tip	White/Slate	P2-21	P4-21	
	Ring	Slate/White	P2-22	P4-22	
54	Tip	Red/Blue	P2-23	P4-23	
	Ring	Blue/Red	P2-24	P4-24	
55	Tip	Red/Orange	P2-25	P4-25	
	Ring	Orange/Red	P2-26	P4-26	
56	Tip	Red/Green	P2-27	P4-27	
	Ring	Green/Red	P2-28	P4-28	
57	Tip	Red/Brown	P2-11	P4-11	
	Ring	Brown/Red	P2-12	P4-12	
58	Tip	Red/Slate	P2-9	P4-9	
	Ring	Slate/Red	P2-10	P4-10	
59	Tip	Black/Blue	P2-7	P4-7	
	Ring	Blue/Black	P2-8	P4-8	
60	Tip	Black/Orange	P2-5	P4-5	
	Ring	Orange/Black	P2-6	P4-6	
61	Tip	Black/Green	P2-3	P4-3	
	Ring	Green/Black	P2-4	P4-4	
62	Tip	Black/Brown	P2-1	P4-1	
	Ring	Brown/Black	P2-2	P4-2	
63	Tip	Black/Slate	P2-29	P4-29	
	Ring	slate/Black	P2-30	P4-30	

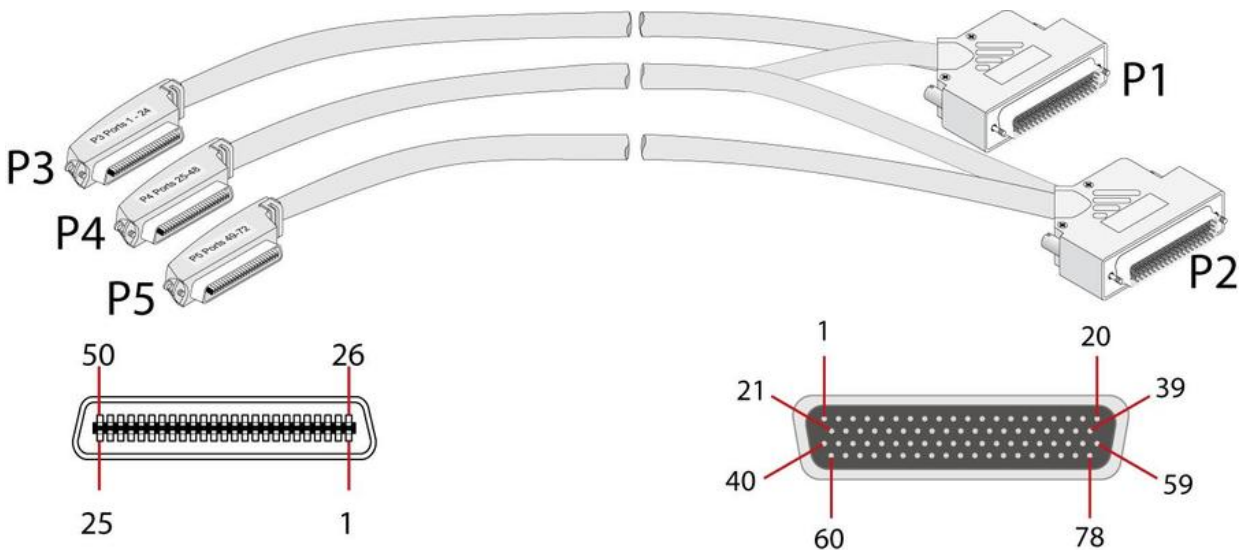
Table 186: dual 78-pin to dual 78-pin cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
64	Tip	Yellow/Blue	P2-61	P4-61	Cable 3 (Continued)
	Ring	Blue/Yellow	P2-60	P4-60	
65	Tip	Yellow/Orange	P2-63	P4-63	
	Ring	Orange/Yellow	P2-62	P4-62	
66	Tip	Yellow/Green	P2-65	P4-65	
	Ring	Green/Yellow	P2-64	P4-64	
67	Tip	Yellow/Brown	P2-67	P4-67	
	Ring	Brown/Yellow	P2-66	P4-66	
68	Tip	Yellow/Slate	P2-41	P4-41	
	Ring	Slate/Yellow	P2-40	P4-40	
69	Tip	Violet/Blue	P2-43	P4-43	
	Ring	Blue/Violet	P2-42	P4-42	
70	Tip	Violet/Orange	P2-45	P4-45	
	Ring	Orange/Violet	P2-44	P4-44	
71	Tip	Black/Orange	P2-47	P4-47	
	Ring	Orange/Black	P2-46	P4-46	
72	Tip	Violet/Brown	P2-49	P4-49	
	Ring	Brown/Violet	P2-48	P4-48	

Dual 78-pin to three 50-pin connector cable

[Figure 170](#) shows the dual 78-pin to three 50-pin connectors cables for 72-port POTS card. [Table 187 on page 1392](#) lists variations of these cables. [Table 188 on page 1394](#) lists pinouts of these cables.

Figure 170: dual 78-pin to three 50-pin cable



Variations of dual 78-pin to three 50-pin connector cables

Table 187 lists variations of the dual 78-pin to three 50-pin connector cables.

Table 187: Variations of dual 78-pin to three 50-pin connector cables

MXK CABLE PART NAME	DESCRIPTION
MXK-CBL-72-CAT3-4FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 4FT/1.22M
MXK-CBL-72-CAT3-10FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 10FT/3.05M
MXK-CBL-72-CAT3-15FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 15FT/4.57M
MXK-CBL-72-CAT3-30FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 30FT/9.10M
MXK-CBL-72-CAT3-50FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 50FT/15.23M
MXK-CBL-72-CAT3-100FT	CABLE: (2) 78PIN TO (3) 78PIN MALE CONNECTORS, 72-PORT CARDS, CAT3 UNSHIELDED, 100FT/30.47M
MXK-CBL-72-CAT5-4FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 4FT/1.22M
MXK-CBL-72-CAT5-10FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 10FT/3.05M
MXK-CBL-72--CAT5-15FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 15FT/4.57M

Table 187: Variations of dual 78-pin to three 50-pin connector cables (Continued)

MXK CABLE PART NAME	DESCRIPTION
MXK-CBL-72-CAT5-30FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 30FT/9.10M
MXK-CBL-72-CAT5-50FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 50FT/15.23M
MXK-CBL-72-CAT5-100FT	CABLE: (2) 78PIN TO (3) 50PIN MALE CONNECTORS, 72-PORT CARDS, CAT5 SHIELDED, 100FT/30.47M

dual 78-pin to three 50-pin cable or blunt pinouts**Table 188: dual 78-pin to three 50-pins or blunt cable pinouts**

Pair	Signal	Color	From	To	Cable
1	Tip	White/Blue	P1-71	P3-26	Cable 1
	Ring	Blue/White	P1-70	P3-1	
2	Tip	White/Orange	P1-73	P3-27	
	Ring	Orange/White	P1-72	P3-2	
3	Tip	White/Green	P1-75	P3-28	
	Ring	Green/White	P1-74	P3-3	
4	Tip	White/Brown	P1-77	P3-29	
	Ring	Brown/White	P1-76	P3-4	
5	Tip	White/Slate	P1-37	P3-30	
	Ring	Slate/White	P1-38	P3-5	
6	Tip	Red/Blue	P1-35	P3-31	
	Ring	Blue/Red	P1-36	P3-6	
7	Tip	Red/Orange	P1-33	P3-32	
	Ring	Orange/Red	P1-34	P3-7	
8	Tip	Red/Green	P1-31	P3-33	
	Ring	Green/Red	P1-32	P3-8	
9	Tip	Red/Brown	P1-53	P3-34	
	Ring	Brown/Red	P1-52	P3-9	
10	Tip	Red/Slate	P1-55	P3-35	
	Ring	Slate/Red	P1-54	P3-10	
11	Tip	Black/Blue	P1-57	P3-36	
	Ring	Blue/Black	P1-56	P3-11	
12	Tip	Black/Orange	P1-59	P3-37	
	Ring	Orange/Black	P1-58	P3-12	
13	Tip	Black/Green	P1-19	P3-38	
	Ring	Green/Black	P1-20	P3-13	
14	Tip	Black/Brown	P1-17	P3-39	
	Ring	Brown/Black	P1-18	P3-14	

Table 188: dual 78-pin to three 50-pins or blunt cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
15	Tip	Black/Slate	P1-15	P3-40	Cable 1 (Continued)
	Ring	Slate/Black	P1-16	P3-15	
16	Tip	Yellow/Blue	P1-13	P3-41	
	Ring	Blue/Yellow	P1-14	P3-16	
17	Tip	Yellow/Orange	P1-11	P3-42	
	Ring	Orange/Yellow	P1-12	P3-17	
18	Tip	Yellow/Green	P1-9	P3-43	
	Ring	Green/Yellow	P1-10	P3-18	
19	Tip	Yellow/Brown	P1-7	P3-44	
	Ring	Brown/Yellow	P1-8	P3-19	
20	Tip	Yellow/Slate	P1-29	P3-45	
	Ring	Slate/Yellow	P1-30	P3-20	
21	Tip	Violet/Blue	P1-27	P3-46	
	Ring	Blue/Violet	P1-28	P3-21	
22	Tip	Violet/Orange	P1-25	P3-47	
	Ring	Orange/Violet	P1-26	P3-22	
23	Tip	Violet/Green	P1-23	P3-48	
	Ring	Green/Violet	P1-24	P3-23	
24	Tip	Violet/Brown	P1-21	P3-49	
	Ring	Brown/Violet	P1-22	P3-24	

Table 188: dual 78-pin to three 50-pins or blunt cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
25	Tip	White/Blue	P1-5	P4-26	Cable 2
	Ring	Blue/White	P1-6	P4-1	
26	Tip	White/Orange	P1-3	P4-27	
	Ring	Orange/White	P1-4	P4-2	
27	Tip	White/Green	P1-1	P4-28	
	Ring	Green/White	P1-2	P4-3	
28	Tip	White/Brown	P1-41	P4-29	
	Ring	Brown/White	P1-40	P4-4	
29	Tip	White/Slate	P1-67	P4-30	
	Ring	Slate/White	P1-66	P4-5	
30	Tip	Red/Blue	P1-65	P4-31	
	Ring	Blue/Red	P1-64	P4-6	
31	Tip	Red/Orange	P1-63	P4-32	
	Ring	Orange/Red	P1-62	P4-7	
32	Tip	Red/Green	P1-61	P4-33	
	Ring	Green/Red	P1-60	P4-8	
33	Tip	Red/Brown	P1-43	P4-34	
	Ring	Brown/Red	P1-42	P4-9	
34	Tip	Red/Slate	P1-45	P4-35	
	Ring	Slate/Red	P1-44	P4-10	
35	Tip	Black/Blue	P1-47	P4-36	
	Ring	Blue/Black	P1-46	P4-11	
36	Tip	Black/Orange	P1-49	P4-37	
	Ring	Orange/Black	P1-48	P4-12	

Table 188: dual 78-pin to three 50-pins or blunt cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
37	Tip	Black/Green	P2-71	P4-38	Cable 2 (Continued)
	Ring	Green/Black	P2-70	P4-13	
38	Tip	Black/Brown	P2-73	P4-39	
	Ring	Brown/Black	P2-72	P4-14	
39	Tip	Black/Slate	P2-75	P4-40	
	Ring	Slate/Black	P2-74	P4-15	
40	Tip	Yellow/Blue	P2-77	P4-41	
	Ring	Blue/Yellow	P2-76	P4-16	
41	Tip	Yellow/Orange	P2-37	P4-42	
	Ring	Orange/Yellow	P2-38	P4-17	
42	Tip	Yellow/Green	P2-35	P4-43	
	Ring	Green/Yellow	P2-36	P4-18	
43	Tip	Yellow/Brown	P2-33	P4-44	
	Ring	Brown/Yellow	P2-34	P4-19	
44	Tip	Yellow/Slate	P2-31	P4-45	
	Ring	Slate/Yellow	P2-32	P4-20	
45	Tip	Violet/Blue	P2-53	P4-46	
	Ring	Blue/Violet	P2-52	P4-21	
46	Tip	Violet/Orange	P2-55	P4-47	
	Ring	Orange/Violet	P2-54	P4-22	
47	Tip	Violet/Green	P2-57	P4-48	
	Ring	Green/Violet	P2-56	P4-23	
48	Tip	Violet/Brown	P2-59	P4-49	
	Ring	Brown/Violet	P2-58	P4-24	

Table 188: dual 78-pin to three 50-pins or blunt cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
49	Tip	White/Blue	P2-13	P5-26	Cable 3
	Ring	Blue/White	P2-14	P5-1	
50	Tip	White/Orange	P2-15	P5-27	
	Ring	Orange/White	P2-16	P5-2	
51	Tip	White/Green	P2-17	P5-28	
	Ring	Green/White	P2-18	P5-3	
52	Tip	White/Brown	P2-19	P5-29	
	Ring	Brown/White	P2-20	P5-4	
53	Tip	White/Slate	P2-21	P5-30	
	Ring	Slate/White	P2-22	P5-5	
54	Tip	Red/Blue	P2-23	P5-31	
	Ring	Blue/Red	P2-24	P5-6	
55	Tip	Red/Orange	P2-25	P5-32	
	Ring	Orange/Red	P2-26	P5-7	
56	Tip	Red/Green	P2-27	P5-33	
	Ring	Green/Red	P2-28	P5-8	
57	Tip	Red/Brown	P2-11	P5-34	
	Ring	Brown/Red	P2-12	P5-9	
58	Tip	Red/Slate	P2-9	P5-35	
	Ring	Slate/Red	P2-10	P5-10	
59	Tip	Black/Blue	P2-7	P5-36	
	Ring	Blue/Black	P2-8	P5-11	
60	Tip	Black/Orange	P2-5	P5-37	
	Ring	Orange/Black	P2-6	P5-12	
61	Tip	Black/Green	P2-3	P5-38	
	Ring	Green/Black	P2-4	P5-13	
62	Tip	Black/Brown	P2-1	P5-39	
	Ring	Brown/Black	P2-2	P5-14	
63	Tip	Black/Slate	P2-29	P5-40	
	Ring	slate/Black	P2-30	P5-15	

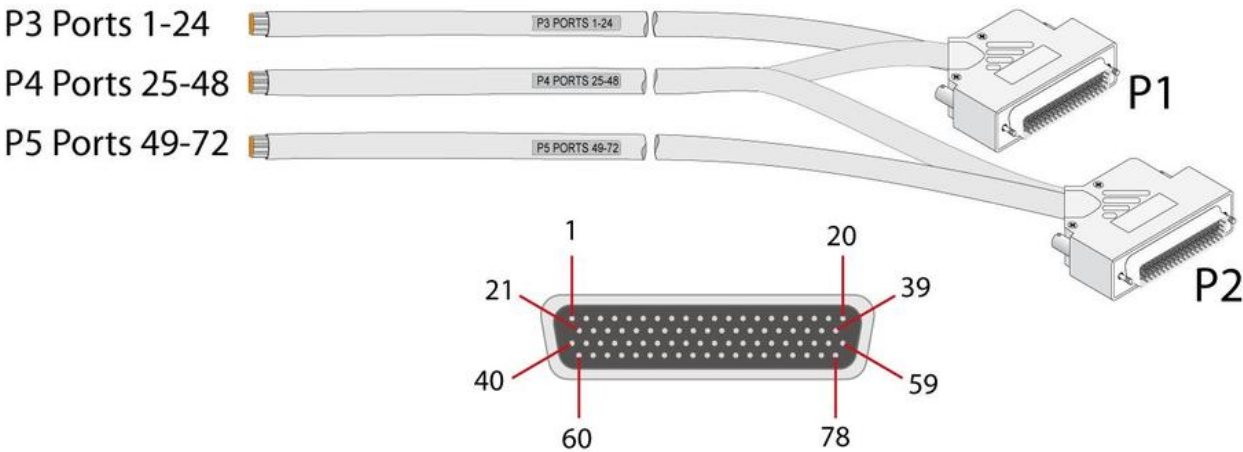
Table 188: dual 78-pin to three 50-pins or blunt cable pinouts (Continued)

Pair	Signal	Color	From	To	Cable
64	Tip	Yellow/Blue	P2-61	P5-41	Cable 3 (Continued)
	Ring	Blue/Yellow	P2-60	P5-16	
65	Tip	Yellow/Orange	P2-63	P5-42	
	Ring	Orange/Yellow	P2-62	P5-17	
66	Tip	Yellow/Green	P2-65	P5-43	
	Ring	Green/Yellow	P2-64	P5-18	
67	Tip	Yellow/Brown	P2-67	P5-44	
	Ring	Brown/Yellow	P2-66	P5-19	
68	Tip	Yellow/Slate	P2-41	P5-45	
	Ring	Slate/Yellow	P2-40	P5-20	
69	Tip	Violet/Blue	P2-43	P5-46	
	Ring	Blue/Violet	P2-42	P5-21	
70	Tip	Violet/Orange	P2-45	P5-47	
	Ring	Orange/Violet	P2-44	P5-22	
71	Tip	Black/Orange	P2-47	P5-48	
	Ring	Orange/Black	P2-46	P5-23	
72	Tip	Violet/Brown	P2-49	P5-49	
	Ring	Brown/Violet	P2-48	P5-24	

Dual 78-pin to blunt connector cable

[Figure 171](#) shows the dual 78-pin to blunt connector cable for 72-port POTS card. [Table 189 on page 1400](#) lists variations of these cables. [Table 188 on page 1394](#) lists pinouts of these cables. Note that the pinouts of two to blunt are same as two to three cables.

Figure 171: dual 78-pin to blunt cable



Variations of dual 78-pin to blunt end cables

Table 189 lists variations of the dual 78-pin to blunt end cables for 72-port POTS card.

Table 189: Variations of dual 78-pin to blunt connector cables

MXK CABLE PART NAME	DESCRIPTION
MXK-CBL-72-CAT3-4FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT3 UNSHIELDED, 4FT/1.22M
MXK-CBL-72-CAT3-10FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT3 UNSHIELDED, 10FT/3.05M
MXK-CBL-72-CAT3-15FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT3 UNSHIELDED, 15FT/4.57M
MXK-CBL-72-CAT3-30FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT3 UNSHIELDED, 30FT/9.10M
MXK-CBL-72-CAT3-50FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT3 UNSHIELDED, 50FT/15.23M
MXK-CBL-72-CAT3-100FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT3 UNSHIELDED, 100FT/30.47M
MXK-CBL-72-CAT5-4FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT5 SHIELDED, 4FT/1.22M
MXK-CBL-72-CAT5-10FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT5 SHIELDED, 10FT/3.05M
MXK-CBL-72-CAT5-15FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT5 SHIELDED, 15FT/4.57M
MXK-CBL-72-CAT5-30FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT5 SHIELDED, 30FT/9.10M

Table 189: Variations of dual 78-pin to blunt connector cables (Continued)

MXK CABLE PART NAME	DESCRIPTION
MXK-CBL-72-CAT5-50FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT5 SHIELDED, 50FT/15.23M
MXK-CBL-72-CAT5-100FT-BLUNT	CABLE: (2) 78PIN TO BLUNT END, 72-PORT CARDS, CAT5 SHIELDED, 100FT/30.47M

15

MXK EFM SHDSL CARDS

This chapter describes the MXK EFM (Ethernet in the First Mile) SHDSL cards: MXK-EFM-SHDSL-24-NTP and MXK-EFM-SHDSL-24-NTWC, G.SHDSL bonding, and configuring G.SHDSL profiles:

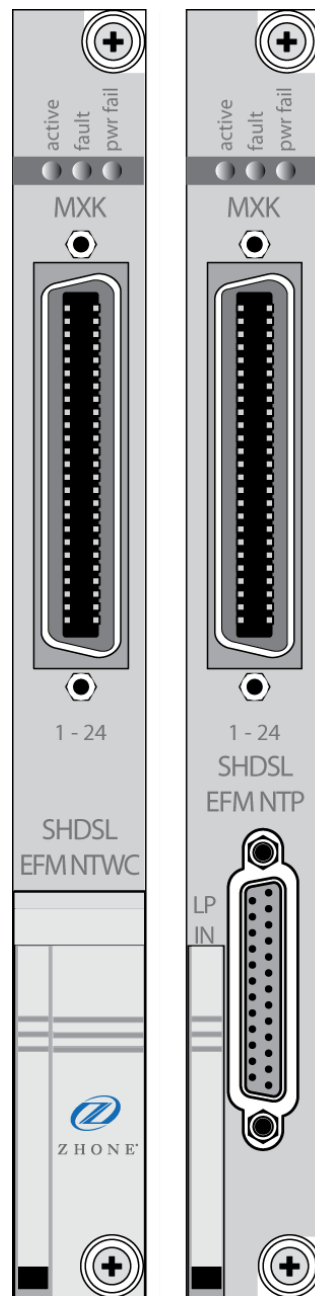
- [EFM SHDSL cards, page 1403](#)
- [MXK EFM SHDSL bonding overview, page 1410](#)
- [G. SHDSL bond group configuration, page 1411](#)
- [Auto-bond type switching, page 1423](#)
- [SNR monitoring for bonded G.SHDSL lines, page 1430](#)
- [SHDSL error monitoring, page 1420](#)
- [SHDSL statistics, page 1452](#)
- [Bond group statistics and port statistics, page 1456](#)
- [EtherXtender statistics, page 1458](#)
- [802.3ah EFM OAM, page 1463](#)
- [MXK-EFM-SHDSL-24 pinouts, page 1465](#)
- [Power and data connections for SHDSL CPE devices, page 1467](#)
- [MTAC testing, page 1470](#)

EFM SHDSL cards

This section describes the MXK EFM SHDSL 24-port cards and how to add them to the system:

- [EFM SHDSL card overview, page 1404](#)
- [EFM SHDSL card specifications, page 1405](#)
- [EFM SHDSL-24 card configuration, page 1406](#)

EFM SHDSL card overview



The MXK SHDSL 24-port cards provide 24 bondable SHDSL ports with a maximum of eight ports per bonded group and a maximum of 24 bonded groups.

This support enables up to three bonded groups of eight ports for 8-port EtherXtend devices, up to six bonded groups of four ports for 4-port EtherXtend devices, and up to 24 groups using one port for each EtherXtend.

The SHDSL line cards used with packet uplink cards, MXK MXK-UPLINK-2X10GE-8X1G, MXK MXK-UPLINK-8X1G, or MXK-UPLINK-4X1G, that support routing and bridging but not cell relay.

The MXK SHDSL 24-port cards provides Ethernet over SHDSL links to Zhone EtherXtends and N2N CPE devices. SHDSL links can be added or removed as the network is configured. The card automatically performs load balancing over the links.

The MXK EMF SHDSL cards are:

- MXK-EFM-SHDSL-24-NTWC

The MXK-EFM-SHDSL-24-NTWC card provides network timing reference and current. The network timing reference allows SHDSL lines to use the backplane clock to clock T1/E1 traffic eliminating the need for a clock source at each location where remote devices are installed.



Note: The legacy Zhone SNE20x0 CPE devices present link stability issues when used with the wetting current version of the EFM-SHDSL line cards (MXK-EFM-SHDSL-24-NTWC). The issue does not exist when the SNE 20x0 are used with the Line Power, No Wetting Current version (MXK-EFM-SHDSL-24-NTP). Zhone recommends using the Line Power versions of the EFM-SHDSL cards for customers planning on deploying SNE 20x0 CPEs.

- MXK-EFM-SHDSL-24-NTP

The MXK-EFM-SHDSL-24-NTP card provides network timing reference and line power. The timing reference enables the card to use the MXK timing as the SHDSL line clocking. This allows an SHDSL CPE to derive timing from the input of the SHDSL lines. It then can use that timing/clocking to provide timing to other subtended devices. The line power feature can be used to power CPEs such as the SkyZhone to eliminate the need for local power. The power is combined with the data and sent out over the 24 SHDSL ports to downstream CPE devices such as the SkyZhone. One MXK-EFM-SHDSL-24-NTP line card can provide power and data for up to 12 CPE devices.

EFM SHDSL card specifications

[Table 190](#) provides the specifications for the MXK-EFM-24 bonding cards.

Table 190: MXK-EFM-SHDSL-24 card specifications

Specification	Description
Density	24 ports
Physical interface	Standard telco connector
Size	1 slot

Table 190: MXK-EFM-SHDSL-24 card specifications (Continued)

Specification	Description
Connectors	One (1) Champ 50-pin telco connector
Line characteristics	ITU G.991.2 SHDSL
Supported line rates	Symmetric rate increments up to 5.7 Mbps
Redundancy	None
Power consumption	34.0 W nominal (all port initialized, no ports trained) plus 0.79 W additional per active SHDSL interface 43.48 W maximum

EFM SHDSL-24 card configuration

This section describes how to configure the EFM SHDSL-24 card:

- [Enter a card-profile for the card, page 1406](#)
- [Set wetting current, page 1408](#)
- [Switch clocking source, page 1409](#)

Enter a card-profile for the card

Each card installed in the system must have a **card-profile**. The type of line card determines the parameter settings in the **card-profile** and the software image for the card. Performing a **card add** automatically creates the **card-profile** for the card with the correct software image and settings.

[Table 191](#) describes the card type and software images for the MXK-EFM-SHDSL-24 cards on the MXK:

Table 191: Card type and software image

Card	Type	Name of software image
MXK-EFM-SHDSL-24-NTP	10208	<i>mxlc24gshdslbond.bin</i>
MXK-EFM-SHDSL-24-NTWC	10208	<i>mxlc24gshdslbond.bin</i>

Creating the card profile for EFM SHDSL-24 cards

Add a SHDSL-24 card to the system.

- 1 Install the SHDSL-24 card in the desired line card slot.
- 2 Create a **card-profile** for the card by entering:

```
zSH> card add 11
new card-profile 1/11/10208 added, sw-file-name "mxlc24gshdslbond.bin"
```

3 Verify the card by entering slots:

```
zSH> slots
MXK 819
```

Uplinks

a:MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)

b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)

Cards

```
4: MXK 4 PORT GPON (RUNNING)
11: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
13: MXK 20 ACT ETH (RUNNING)
```

4 Verify the card-profile for the SHDSL-24 card:

```
zSH> get card-profile 1/11/10208
card-profile 1/11/10208
sw-file-name: -----> {mxlc24gshdslbond.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {unknown type}
card-atm-configuration: -> {not applicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {not applicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}
```

5 View card information including the state of the card and how long the card has been running:

```
zSH> slots 11
MXK 819
Type           : MXK GSHDSL-24 Bonded
Sub-Type       : with NTP
Card Version   : 00001
EEPROM Version : 1
Serial #       : 962646
CLEI Code      : No CLEI
Card-Profile ID : 1/11/10208
Shelf          : 1
Slot           : 11
ROM Version    :
Software Version: MXK 1.16.2.109
```

```

State           : RUNNING
Mode            : FUNCTIONAL
Heartbeat check : enabled
Longest hbeat   : 50
Fault reset     : enabled
Uptime          : 6 minutes

```

Set wetting current

The following example enables the wetting current feature. Wetting current provides 10-15 mA per G.SHDSL line. The default setting for the wetting-current parameter is disabled.



Note: The MXK G.SHDSL card cannot be pre-provisioned with wetting current. First install the card, then create the **card-profile with wetting-current set to standard**. Or pre-provision the **card-profile** without wetting current, then update the **card-profile** to enable wetting current by changing the variable to *standard* after the card is installed.



Note: Enabling wetting current from ZMS causes the card to reboot.

Setting wetting current for the MXK-EFM-SHDSL-24-NTWC card

To enable this feature, change the wetting-current parameter to standard.

```

zSH> update card-profile 1/11/10208
card-profile 1/11/10208
Please provide the following: [q]uit.
sw-file-name: -----> {mxlc24gshdslbond.bin}:
admin-status: -----> {operational}:
upgrade-sw-file-name: ---> {}:
upgrade-vers: -----> {}:
admin-status-enable: ----> {enable}:
sw-upgrade-admin: -----> {reloadcurrrev}:
sw-enable: -----> {true}:
sw-upgrade-enable: -----> {false}:
card-group-id: -----> {0}:
hold-active: -----> {false}:
weight: -----> {nopreference}:
card-line-type: -----> {unknowntype}:
card-atm-configuration: -> {notapplicable}:
card-line-voltage: -----> {not-used}:
maxvpi-maxvci: -----> {notapplicable}:
card-init-string: -----> {}:
wetting-current: -----> {disabled}: standard
pwe-timing-mode: -----> {none}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s

```

Record updated.

Switch clocking source

The default clock source setting is **ntr-local-osc**. To change the default setting for the NTR setting, enter the parameter **ntr-refck-8khz**. The clock source will switch from the local oscillator to the backplane 8Kz reference clock. This affects all ports on the card.

Switching clocking for the MXK-EFM-SHDSL-24-NT/NTP cards

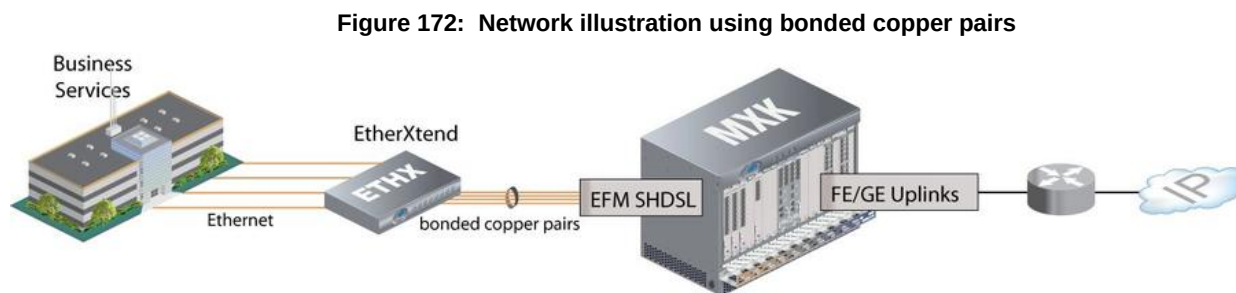
Enter **ntr-refck-8khz** in the **efmCuPmeNtr** parameter field.

```
zSH> update pme-profile 1-11-1-0/shdsl
pme-profile 1-11-1-0/shdsl
Please provide the following: [q]uit.
efmCuPmeAdminSubType: -----> {ieee2baset1r}:
efmCuPmeAdminProfile: -----> {0}:
efmCuPAFRemoteDiscoveryCode: -----> {}:
efmCuPmeThreshLineAtn: -----> {0}:
efmCuPmeThreshMinSnrMgn: -----> {0}:
efmCuPmeLineAtnCrossingEnable: -----> {false}:
efmCuPmeSnrMgnCrossingTrapEnable: ---> {false}:
efmCuPmeDeviceFaultEnable: -----> {false}:
efmCuPmeConfigInitFailEnable: -----> {false}:
efmCuPmeProtocolInitFailEnable: -----> {false}:
efmCuPme2BProfileDescr: -----> {}:
efmCuPme2BRegion: -----> {region1}:
efmCuPme2BDataRate: -----> {0}:
efmCuPme2BPower: -----> {0}:
efmCuPme2BConstellation: -----> {adaptive}:
efmCuPme2BProfileRowStatus: -----> {active}:
efmCuPmeNtr: -----> {ntr-local-osc}: ntr-refck-8khz
efmCuPmeThreshMaxSnrMgnDelta: -----> {20}:
efmCuPmeMaintenanceMode: -----> {off}:
efmCuPmeMaintenanceStartTime: -----> {00:00}:
efmCuPmeMaintenanceEndTime: -----> {23:59}:
efmCuPmeSnrMonitoringInterval: -----> {01:00}:
efmCuPmeErrorThreshMonEnable: -----> {false}:
efmCuPmeErrorThreshMonNotifyEnable: -> {false}:
efmCuPmeErrorThreshMonInterval: -----> {12}:
efmCuPmeErrorThreshMonClrInterval: --> {1800}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

MXK EFM SHDSL bonding overview

MXK EFM (Ethernet in the first Mile) interfaces provide high speed, symmetrical Ethernet services over SHDSL links between the MXK and network extender EFM modems and Ethernet access devices like Zhone's EtherXtend and N2N CPE devices.

Figure 172 shows a typical network scenario using bonded copper pairs.



G. SHDSL bond group configuration

This section describes:

- [Bond group bandwidth specifications, page 1411](#)
- [Bond group configuration, page 1412](#)
- [View bond groups, page 1416](#)
- [Change bond group type, page 1417](#)
- [Move bond group members, page 1418](#)
- [Delete bond groups, page 1419](#)
- [SHDSL error monitoring, page 1420](#)

MXK G.SHDSL interfaces support 802.3ah EFM standards and Zhone's proprietary Ethernet bonding technologies. MXK Ethernet bonding supports up to 24 bond groups with a maximum of eight ports in a bond group. G.SHDSL links can be added or removed from bond groups as the network is configured. The MXK automatically performs load balancing over the links.

Conditions and limitations for cross-card bonding

EFM cross-card bonding on the MXK has the following conditions and limitations:

- Cross-card bonding is not supported in N2N mode.
- Bond groups span a maximum of two cards.
- The primary card can handle a maximum of 48 ports.
- Bond groups can be created from both older and newer line cards.

Bond group bandwidth specifications

[Table 192](#) shows the bond group bandwidth rates for EFM bond groups with four ports.

Table 192: Bond group bandwidth for four-port bond groups

Frame size	Downstream (pkts/sec)	Upstream (pkt/sec)	Total
64	40584	40584	81168
128	21748	21748	42956
256	11105	11105	22210
512	5547	5547	11094
1024	2826	2826	5652
1280	2269	2269	4538

Table 192: Bond group bandwidth for four-port bond groups (Continued)

Frame size	Downstream (pkts/sec)	Upstream (pkt/sec)	Total
1480	1967	1967	3934

Bond group configuration

This section describes:

- [EFM auto bonding, page 1412](#)
- [EFM manual bond groups, page 1414](#)
- [Create bond groups on one card, page 1414](#)

EFM auto bonding

EFM discovery automatically groups ports that are connected to the same CPE to create a dynamic bond group utilizing automatic creation bond group numbers.

Bond group numbering is from 25 to 505. Automatic bonding, as well as CLI and ZMS bond group creation, create bond group numbers in this range.

The **efmCuPAFAutoDiscovery** parameter variables in the **efm-port** profile on the MXK (the CO side of the configuration) determine how EFM bond groups and their members are created.

```
efmCuPAFAutoDiscovery:-----> disabled optional required
```

[Table 193](#) defines the variables for the **efmCuPAFAutoDiscovery** parameter.

Table 193: efmCUPAFAutoDiscovery parameter

Parameter	Description
efmCuPAFAutoDiscovery	<i>disabled</i>: EFM bond groups and the members must be manually provisioned. <i>optional</i>: EFM bond groups and members may be manually or automatically provisioned. If the links were manually added to a bond group, this configuration is used and EFM Discovery is not performed. If the links were not manually added to a bond group and the CPE supports EFM Discovery, provisioning will be automatic. <i>required</i>: EFM bond groups and members are automatically provisioned with the EFM Discovery algorithm. If the CPE device does not support EFM Discovery, the link will not be obtained. Default: <i>optional</i>

EFM SHDSL cards on the MXK support up to 24 bond groups. Each bond group can have a maximum of eight members.

The number of bond groups on a SHDSL card depend on the number of ports that exist on the CPE devices connected to the EFM SHDSL card. For example, a EFM SHDSL card connected to six four-port CPE devices would have six bond groups.

Discovery works only with a discovery capable CPE device and when the EFM SHDSL **efmCuPAFAutoDiscovery** parameter is set to either *required* or *optional*.

When the parameter is set to *optional* and the CPE is not discovery capable, a dynamic bond group will not be created, but the link will remain. If the CPE is discovery capable, a dynamic bond group is created with the port as a member of the bond group. If the port already belongs to a bond group, the bond group type will be checked. If the bond group type is *dynamic*, the discovery messages are sent to verify that the ports are still connected to the same CPE. If the bond group type was created from the CLI or with ZMS, (25-99 or 101-199), discovery messages are not sent and the current configuration of bond groups remain as they are.

When the parameter is set to *required*, the CPE must be discovery capable or the links will fail. When *required* is set, discovery automatically creates bond groups depending on the number of CPE ports connected to the EFM SHDSL card.

Updating efm-port for auto-bonding

The **efmCuPAFAutoDiscovery** parameter in the **efm-port** profile must be changed to *required* or *optional* to enable auto-bonding.



Caution: The **efmCuPAFAutoDiscovery** parameter *must* have the same setting for all ports destined to the same bond group.

It is an unenforced illegal setting to have the CO **efm-port(s)** connected to the same CPE (destined to the same bond group) configured with different discovery settings.

When a configuration with different settings exists, those ports configured as *required* or *optional* will be put into one bond group, and those ports configured with disabled will either go to a different bond group or none at all causing a mismatch in traffic flow.

- 1 View the ports on the CO side that will be connected to the CPE device. The default for the **efmCuPAFAutoDiscovery** parameter is *optional* to enable *autodiscovery*.

```
zSH> get efm-port 1-4-1-0/shdsl
efm-port 1-4-1-0/shdsl
efmCuPAFAdminState: -----> {enabled}
efmCuPAFDiscoveryCode: -----> {}
efmCuAdminProfile: -----> {0x01}
efmCuTargetDataRate: -----> {50000}
```

```
efmCuTargetWorstCaseSnrMgn: -----> {0}
efmCuThreshLowBandwidth: -----> {0}
efmCuLowBandwidthEnable: -----> {false}
efmCuTargetCurrentConditionMode: ---> {false}
efmCuTargetCurrentConditionSnrMgn: -> {6}
efmCuTargetWorstCaseMode: -----> {true}
efmCuPAFAutoDiscovery: -----> {optional}
```

2 View the bond group automatically created.

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
4	26	efmbond	ACT	1-4-26-0	-
4	25	efmbond	ACT	1-4-25-0	-
4	299	efmbond	ACT	bond-0299	-

3 View the links in the automatically created bond group.

```
zSH> bond show group bond-0299/efmbond
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
4	299	efmbond	ACT	bond-0299	-

Group Members					
Slot	Port	Type	State	Name	Desc
4	3	shdsl	ACT	1-4-3-0	-
4	1	shdsl	ACT	1-4-1-0	-
4	7	shdsl	ACT	1-4-7-0	-
4	2	shdsl	ACT	1-4-2-0	-
4	6	shdsl	ACT	1-4-6-0	-
4	8	shdsl	ACT	1-4-8-0	-
4	5	shdsl	ACT	1-4-5-0	-
4	4	shdsl	ACT	1-4-4-0	-

EFM manual bond groups

When you are not using EFM discovery and want to create bond groups from the CLI, first create the EFM bond group, then add the links to that group before connecting the CPE.

To switch from EFM to N2N bonding, the EFM links can be moved from the bond group to a N2N bond group or disconnect the CPE and delete the EFM bond group.

The MXK SHDSL connector has 24 SHDSL ports and supports up to 24 bond groups.

Create bond groups on one card

Creating bond groups on one card

Creating a bond group requires that you first create the bond group, then add one or more bond group members.

- 1 Enter **slots** to verify the location of the EFM G.SHDSL card.

```
zSH> slots
MXK 819
Uplinks
  a:*MXK EIGHT GIGE (RUNNING+TRAFFIC)
  b: MXK EIGHT GIGE (RUNNING)
Cards
  1: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
  2: MXK GSHDSL-24 Bonded/with NTWC (RUNNING)
  4: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
```

- 2 Enter **bond add group shelf-slot-port-subport/type** to create a bond group with *port* designating the group ID of the bond group.



Note: In the case of bond group commands, *port* refers to the group ID of the bond group.

When entering a bond group *interface/efmbond*, check to see which *interface* is actually created. If the bond group already exists, the system creates the interface with a system assigned value. For example,

```
zSH> bond add group 1-1-30-0/efmbond
Bond group - bond-0030/efmbond - was successfully created.
```



Note: Note that the bond group created has a different *interface* than the *interface* entered.

- 3 Verify the bond group name:

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
1	30	efmbond	OOS	bond-0030	-
4	26	efmbond	ACT	1-4-26-0	-
4	25	efmbond	ACT	1-4-25-0	-
4	299	efmbond	ACT	bond-0299	-

- 4 Enter **bond show group interface/type** to verify the bond group.

```
zSH> bond show group bond-0030/efmbond
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
1	30	efmbond	OOS	bond-0030	-

- 5 Add a single bond group member to the bond group by entering **bond add member shelf-slot-port-subport/type shelf-slot-port-subport/type** with the interface and type of the bond group followed by the interface and type of the group member to be added.

```
zSH> bond add member bond-0030/efmbond 1-1-1-0/shdsl
```

Add several bond group members to the bond group:

```
zSH> bond add member bond-0030/efmbond 1-1-2-0/shdsl 1-1-3-0/shdsl 1-1-4-0/
shdsl 1-1-5-0/shdsl 1-1-6-0/shdsl 1-1-7-0/shdsl 1-1-8-0/shdsl
```

6 Enter **bond show group shelf-slot-port-subport/type** to verify the members of the bond group.

```
zSH> bond show group bond-0030/efmbond
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
1	30	efmbond	OOS	bond-0030	-

Group Members					
Slot	Port	Type	State	Name	Desc
1	1	shdsl	OOS	1-1-1-0	-
1	8	shdsl	OOS	1-1-8-0	-
1	7	shdsl	OOS	1-1-7-0	-
1	6	shdsl	OOS	1-1-6-0	-
1	5	shdsl	OOS	1-1-5-0	-
1	4	shdsl	OOS	1-1-4-0	-
1	3	shdsl	OOS	1-1-3-0	-
1	2	shdsl	OOS	1-1-2-0	-

View bond groups

You can view all existing bond groups, a specific bond group, all the bond groups on a specific slot, or view the bond group of a specific link.

Viewing all bond groups

View all configured bond groups.

Enter **bond show all** to view all bond groups.

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
1	30	efmbond	OOS	bond-0030	-
4	26	efmbond	ACT	1-4-26-0	-
4	25	efmbond	ACT	1-4-25-0	-
4	299	efmbond	ACT	bond-0299	-

Viewing a specific bond group

View a specific bond group.

Enter **bond show group shelf-slot-port-subport/type** to view the bond group and all of members of the bond group:

```
zSH> bond show group bond-0030/efmbond
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
1	30	efmbond	OOS	bond-0030	-

Group Members					
Slot	Port	Type	State	Name	Desc
1	1	shdsl	OOS	1-1-1-0	-

1	8	shdsl	OOS	1-1-8-0	-
1	7	shdsl	OOS	1-1-7-0	-
1	6	shdsl	OOS	1-1-6-0	-
1	5	shdsl	OOS	1-1-5-0	-
1	4	shdsl	OOS	1-1-4-0	-
1	3	shdsl	OOS	1-1-3-0	-
1	2	shdsl	OOS	1-1-2-0	-

Viewing bond groups by slot

View bond groups by slot.

Enter **bond show slot slotnumber** to view the bond groups.

```
zSH> bond show slot 1
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
1	30	efmbond	OOS	bond-0030	-

Viewing the bond group of a specific link

View the bond group of a specific link.

Enter **bond show link shelf-slot-port-subport/type** to view the bond group of a specific link and any other links in the group:

```
zSH> bond show link 1-1-8-0/shdsl
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
1	30	efmbond	OOS	bond-0030	-
Group Members					
Slot	Port	Type	State	Name	Desc
1	1	shdsl	OOS	1-1-1-0	-
1	8	shdsl	OOS	1-1-8-0	-
1	7	shdsl	OOS	1-1-7-0	-
1	6	shdsl	OOS	1-1-6-0	-
1	5	shdsl	OOS	1-1-5-0	-
1	4	shdsl	OOS	1-1-4-0	-
1	3	shdsl	OOS	1-1-3-0	-
1	2	shdsl	OOS	1-1-2-0	-

Change bond group type

The bond group type can be changed for individual bond groups or all bond groups used in a specified slot using the **bond modify** commands.

Changing one bond group type

Change one bond group type.

- 1 Enter **bond modify type group shelf-slot-port-subport/type** to change a bond group from *efm* to *n2n*:

```
zSH> bond modify n2n group bond-0030/efmbond
```

2 Enter **bond show all** to view the change in type.

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
1	30	n2nbond	OOS	bond-0030	-

Changing all the bond group types on a device

To change all of the bond groups on the MXK in slot 1 from EFM to N2N enter:

1 View the bond group and their type.

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
1	28	n2nbond	OOS	bond-0028	-
1	27	n2nbond	OOS	bond-0027	-

2 Enter **bond modify type slot slot number** to change the bonding type, in this case from n2n to efm.

```
zSH> bond modify efm slot 1
```

3 Enter **bond show all** to view the changes.

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
1	28	efmbond	OOS	bond-0028	-
1	27	efmbond	OOS	bond-0027	-

Move bond group members

Bond group members can be moved from one group to another, even between bond groups of different types.

Moving bond group members

If a bond group is full, you will not be allowed to complete the move.

Enter **bond show all** to view the bond groups and their types.

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
1	28	efmbond	OOS	bond-0028	-
1	27	efmbond	OOS	bond-0027	-

Enter **bond move name/type name/type shelf-slot-port-subport/type** to move a link from one group to another.

```
zSH> bond move bond-0028/efmbond bond-0027/efmbond 1-1-9-0/shdsl
```

```
zSH> bond move bond-0028/efmbond bond-0027/efmbond 1-1-10-0/shdsl
```

Caution: group bond-0028/efmbond is now empty!

The system notifies you when the bond group is empty.

Enter **bond show link** *shelf-slot-port-subport/type* to view the bond group where to member was moved to.

```
zSH> bond show link 1-1-9-0/shdsl
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
1	27	efmbond	OOS	bond-0027	-
Group Members					
Slot	Port	Type	State	Name	Desc
1	9	shdsl	OOS	1-1-9-0	-
1	10	shdsl	OOS	1-1-10-0	-

Delete a member of the bond group:

Enter **bond delete member** *name/type shelf-slot-port-subport/type* to remove a member of a bond group.

```
zSH> bond delete member bond-0027/efmbond 1-1-9-0/shdsl
```

Delete bond groups

Bond groups can be deleted by individual member or entire group.

Deleting bond groups

Enter **bond show all** to view current bond groups, delete a bond group, and verify that the bond group is deleted.

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
11	27	efmbond	OOS	1-11-27-0	-
11	26	efmbond	OOS	1-11-26-0	-
11	25	efmbond	OOS	1-11-25-0	-

Enter **bond delete group** *name/type* to delete a bond group.

```
zSH> bond delete group 1-11-26-0/efmbond
```

Enter **bond show all** to verify the deletion.

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
11	27	efmbond	OOS	1-11-27-0	-
11	25	efmbond	OOS	1-11-25-0	-

Cross-card bonding

Creating bond groups across cards (cross-card bonding)

Creating a bond group across cards requires that you first create the bond group, then add one or more bond group members from two cards.

1 Create the bond group.

```
zSH> bond add group 1-1-27-0/efmbond
Bond group - bond-0027/efmbond - was successfully created.
```

2 View the bond group.

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
1	27	efmbond	OOS	bond-0027	-

3 Add bond group members from two cards.

```
zSH> bond add member bond-0027/efmbond 1-1-1-0/shdsl 1-1-2-0/shdsl 1-1-3-0/
shdsl 1-1-4-0/shdsl 1-2-1-0/shdsl 1-2-2-0/shdsl 1-2-3-0/shdsl 1-2-4-0/shdsl
```

4 Verify bond group members.

```
zSH> bond show group bond-0027/efmbond
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
1	27	efmbond	OOS	bond-0027	-

Group Members					
Slot	Port	Type	State	Name	Desc
1	1	shdsl	OOS	1-1-1-0	-
2	4	shdsl	OOS	1-2-4-0	-
2	3	shdsl	OOS	1-2-3-0	-
2	2	shdsl	OOS	1-2-2-0	-
2	1	shdsl	OOS	1-2-1-0	-
1	4	shdsl	OOS	1-1-4-0	-
1	3	shdsl	OOS	1-1-3-0	-
1	2	shdsl	OOS	1-1-2-0	-

SHDSL error monitoring

This section describes:

- [SHDSL error monitoring statistics, page 1421](#)
- [SHDSL error monitoring fields, page 1421](#)

The MXK provides the **errmon** command to view SHDSL error monitoring information.

SHDSL error monitoring statistics

Enter the **errmon stats** *interface/type* command to view SHDSL error monitoring statistics:

```
zSH> errmon stats 1-4-3-0/shdsl
      Shdsl Error Monitoring Stats
Port  TC Down    CRC    ES    SES    Max Err Sec  Restart  Line Status
3      0        0     61    61      1         0        ACT
```

[Table 194](#) defines the **errmon stats** fields.

Table 194: Definitions of displayed statistics

Parameter	Description
TC Down	Count of how many times the TC layer went down since the physical link was obtained.
CRC	Count of CRC anomalies.
ES	Count of one second intervals during which one or more CRCs are reported.
SES	Count of one second intervals during which at least 50 CRCs are reported.
Max Errored Sec	Maximum consecutive seconds with errors without causing action to be taken by errmon features.
Restart	Count of the number of times the port was restarted by errmon features.
Line Status	Current status of the port.

SHDSL error monitoring fields

The **errmon** *interface/type* command monitors the physical interface, for example, *1-5-1-0/shdsl*.

Enter **errmon show** to view the SHDSL error monitoring fields:

```
zSH> errmon show 1-4-3-0/shdsl
      Shdsl Error Monitoring
Port  Monitor  Notify  ErrorInt  ClearInt  Line Status
3     disabled disabled    12      1800        ACT
```

The *Monitor* and the *Notify* fields must be enabled from the CLI:

```
zSH> errmon modify 1-4-3-0/shdsl monitor enable errmon show
1-4-3-0/shdsl
      Shdsl Error Monitoring
Port  Monitor  Notify  ErrorInt  ClearInt  Line Status
3     enabled disabled    12      1800        ACT

zSH> errmon modify 1-4-3-0/shdsl notify enable errmon show 1-4-3-0/shdsl
      Shdsl Error Monitoring
```

Port	Monitor	Notify	ErrorInt	ClearInt	Line Status
3	enabled	enabled	12	1800	ACT

Table 195: errmon parameters

Parameter	Description
link interface (port)	Name and type of a physical interface. For example, <i>1-5-1-0/shdsl</i> .
monitor	Determines if error monitoring should be performed on the link.
notify	Determines if a notification to the CLI, alarm manager, and ZMS should be generated when an error threshold is exceeded or cleared on the link.
errinterval	Specifies the number of consecutive seconds of detecting errors that, once reached, causes the physical line to be considered a poor performer and action to be taken.
clrinterval	Specifies the number of consecutive error-free seconds that must be achieved in order to declare the physical line usable for transporting data traffic.

errmon modify

Syntax `error modify|show|stats link/interface`

Auto-bond type switching

MXK Ethernet bonding supports bond group monitoring and automatic switching from one bond group type to another bond group type.

The rules for bond group switching are:

- Only configured bond groups are monitored whether they were created automatically or manually.
- EFM cross-card bond groups cannot reconfigure to N2N.
- Monitoring begins when a bond-group enters the “all links down” state.

The **efmCuPAFAutoDiscovery** parameter in the **efm-profile** profile must be set to either *optional* or *required* for all links in the bond group to enable bond group monitoring over system reboots. The bond group automatically switches from one type to another type as the links come up after the “all links down” state. This occurs when the CPE type is changed and auto bond detect is enabled. Auto detection is persistent and continues across reboots.

Enabling or disabling auto detection on the MXK

Enable bond group monitoring and automatic switching from one bond group type to another bond group type by entering the **bondautodetect** command on the MXK.

- 1 Enter the **bondautodetect enable** command to enable auto detection of N2N and EFM bond types.

```
zSH> bondautodetect enable
Bond Auto Detect status:  ON
```

The system can now attempt link in either N2N or EFM bond mode at 300 second (five minute) intervals.

- 2 If necessary, enter the **bondautodetect disable** command to terminate auto detection.

```
zSH> bondautodetect disable
Bond Auto Detect status:  OFF
```

Configure the pme-profile

A **pme-profile** (Physical Medium Entities) is available for each G.SHDSL port. PME profiles are used to set link rates.

To display the PME parameters in their default state, enter **get pme-profile shelf-slot-port-subport/type**:

```
zSH> get pme-profile 1-1-1-0/shdsl
pme-profile 1-1-1-0/shdsl
efmCuPmeAdminSubType: -----> {ieee2baset1r}
efmCuPmeAdminProfile: -----> {0}
```

```

efmCuPAFRemoteDiscoveryCode: -----> {}
efmCuPmeThreshLineAtn: -----> {0}
efmCuPmeThreshMinSnrMgn: -----> {0}
efmCuPmeLineAtnCrossingEnable: -----> {false}
efmCuPmeSnrMgnCrossingTrapEnable: ---> {false}
efmCuPmeDeviceFaultEnable: -----> {false}
efmCuPmeConfigInitFailEnable: -----> {false}
efmCuPmeProtocolInitFailEnable: -----> {false}
efmCuPme2BProfileDescr: -----> {}
efmCuPme2BRegion: -----> {region1}
efmCuPme2BDataRate: -----> {0}
efmCuPme2BPower: -----> {0}
efmCuPme2BConstellation: -----> {adaptive}
efmCuPme2BProfileRowStatus: -----> {active}
efmCuPmeNtr: -----> {ntr-local-osc}
efmCuPmeThreshMaxSnrMgnDelta: -----> {20}
efmCuPmeMaintenanceMode: -----> {off}
efmCuPmeMaintenanceStartTime: -----> {00:00}
efmCuPmeMaintenanceEndTime: -----> {23:59}
efmCuPmeSnrMonitoringInterval: -----> {01:00}
efmCuPmeErrorThreshMonEnable: -----> {false}
efmCuPmeErrorThreshMonNotifyEnable: -> {false}
efmCuPmeErrorThreshMonInterval: -----> {12}
efmCuPmeErrorThreshMonClrInterval: --> {1800}

```

Configure automatic baud rate adaption and fixed rate settings

The line type for an G.SHDSL interface on the MXK is set at **shdsl-2btl** which can perform automatic baud rate adaption. This allows receiving devices to communicate with transmitting devices operating at different baud rates without the need to establish data rates in advance. By determining the baud rate from the transmitting device, the receiving MXK automatically trains to match the line rate of the incoming data.

The automatic baud rate adaption process may take several minutes. This is because the MXK (co) and CPE devices use an algorithm to step through a sequence of baud rates, where the devices establish a connection at each line rate and then move to the next higher rate until they reach the final rate they agree upon.

For the **efmCuPme2BDataRate** parameter, setting the parameter to 0 sets the data rate to adaptive. Entering a value defines a specific data rate in kbps.

```
efmCuPme2BDataRate:-----> {0 - 15352}
```

Before adjusting the data rate in the **efmCuPmeDataRate** parameter, see [Table 196](#) to understand the implications of setting data rates for both *co* and *cpe* mode. [Table 197](#) describes the adaptive [fixed-rate=0] and fixed line rate settings defined in the **efmCuPme2BDataRate** entry of the **pme-profile**.

Table 196: Fix-bit-rate settings and modem train rates

CO	CPE	Then
efmCuPme2BDataRate = 0	efmCuPme2BDataRate = 0	Highest available rate is negotiated.
efmCuPme2BDataRate = 0	efmCuPme2BDataRate = non-zero value = x	x is treated as the MAXIMUM train rate allowed. Modems train at x or less.
efmCuPme2BDataRate = non-zero value = x	efmCuPme2BDataRate = 0	x is treated as the MAXIMUM train rate allowed. Modems train at x or less.
efmCuPme2BDataRate = non-zero value = x	efmCuPme2BDataRate = non-zero-value = x	x is treated as the MAXIMUM train rate allowed. Modems train at x or less.
efmCuPme2BDataRate = non-zero value = x	efmCuPme2BDataRate = non-zero-value = y	The lesser of x and y = z will be treated as the MAXIMUM train rate allowed. Modems train at z or less.

Configure auto-negotiate or specific data rate

For the **efmCuPme2BDataRate** parameter, setting the parameter to 0 sets the data rate to *adaptive*. Entering a data rate value defines a specific data rate that must correspond with the constellation set in the **efmCuPme2BConstellation** parameter (see [Table 197](#)).

For TCPAM16 and TCPAM32 you can set the **efmCuPme2BDataRate** either to adaptive (0) or a data rate corresponding to the TCPAM setting.



Note: When TCPAM16 and TCPAM32 are set to adaptive (0), the maximum data rate is always 5696 kbps on each line configured as adaptive.

For TCPAM4, TCPAM8, and TCPAM64, a data rate *must* be set in the **efmCuPme2BDataRate** parameter. The adaptive rate of 0 (auto-negotiate) cannot be used for these constellation settings. See [Table 197](#) for data rate ranges.



Note: Configuring TCPAM is necessary only on the CO side. However, the CPE must be capable of extended rates in order for the line to successfully train/link.

TCPAM settings are typically used as follows:

- TCPAM4 and TCPAM8 are useful for operations over longer and/or noisier loops, but use reduced data rates.

- TCPAM64 is useful on short loops using CAT5 wiring for in-building and campus applications. In these cases, CAT3 loops would be too noisy for the SNR requirements.



Note: When configuring the MXK for TCPAM 64, the CAT5 rating needs to apply to all intermediary connection points and accessories such as:

- 66-blocks/110-blocks
- jumper wires used on cross-connect blocks, protector blocks, etc.

It is also recommended that the number of RJ21/AMP connectors used between the DSLAM and the final cable are kept to a minimum. It is also recommended that you use CAT5 punch-down blocks as opposed to those blocks that are pre-terminated with AMP connectors.

Configure constellation for a TCPAM setting

TCPAM configurations have a minimum and a maximum data rate range, depending on the constellation setting. If a data rate is set outside of the correct range of the constellation setting, an error message will return.

Table 197 provides the settings for the **efmCuPme2BConstellation** parameter and the rate range for the **efmCuPme2BDataRate** parameter.

When the train rate is set to *adaptive* mode, the specific value configured in the **efmCuPme2BDataRate** field becomes the maximum rate. This means that the line will train successfully to any train rate at or below the configured value and will train at the specified rate unless line conditions dictate otherwise.

When the constellation is not *adaptive* (i.e. is set to *tcpam4*, *tcpam8*, *tcpam16*, *tcpam32*, or *tcpam64*), the specific value set in the **efmCuPme2BDataRate** field becomes fixed. This means that if the requested data rate cannot be achieved due to line conditions, the line will not get link.



Note: N2N bonding supports TCPAM16/32, but not TCPAM4/8/64.

Table 197: efmCuPme2BConstellation and efmCuPme2BDataRate parameter settings

efmCuPme2BConstellation settings	efmCuPme2BData range rates
TCPAM4	192 to 2048
TCPAM8	192 to 5056
TCPAM16	192 to 7616 or an adaptive rate of 0 with a max train rate of 5696
TCPAM32	768 to 10176 or an adaptive rate of 0 with a max train rate of 5696
TCPAM64	768 to 12736

Table 197: efmCuPme2BConstellation and efmCuPme2BDataRate parameter settings (Continued)

efmCuPme2BConstellation settings	efmCuPme2BData range rates
Adaptive	0 or 192 to 5696

Configuring the constellation for a TCPAM setting

Update the **pme-profile** for TCPAM and data rates.

This example sets the **efmCuPme2BDataRate** parameter to *12736* and changes the **efmCuPme2BConstellation** parameter to *tcpam64*.

```

zSH> update pme-profile 1/1/1/0/shdsl
pme-profile 1/1/1/0/shdsl
Please provide the following: [q]uit.
efmCuPmeAdminSubType: -----> {ieee2basetlr}:
efmCuPmeAdminProfile: -----> {0}:
efmCuPAFRemoteDiscoveryCode: -----> {}:
efmCuPmeThreshLineAtn: -----> {0}:
efmCuPmeThreshMinSnrMgn: -----> {0}:
efmCuPmeLineAtnCrossingEnable: -----> {false}:
efmCuPmeSnrMgnCrossingTrapEnable: ---> {false}:
efmCuPmeDeviceFaultEnable: -----> {false}:
efmCuPmeConfigInitFailEnable: -----> {false}:
efmCuPmeProtocolInitFailEnable: -----> {false}:
efmCuPme2BProfileDescr: -----> {}:
efmCuPme2BRegion: -----> {region1}:
efmCuPme2BDataRate: -----> {0}: 12736
efmCuPme2BPower: -----> {0}:
efmCuPme2BConstellation: -----> {adaptive}: tcpam64
efmCuPme2BProfileRowStatus: -----> {active}:
efmCuPmeNtr: -----> {ntr-local-osc}:
efmCuPmeThreshMaxSnrMgnDelta: -----> {20}:
efmCuPmeMaintenanceMode: -----> {off}:
efmCuPmeMaintenanceStartTime: -----> {00:00}:
efmCuPmeMaintenanceEndTime: -----> {23:59}:
efmCuPmeSnrMonitoringInterval: -----> {01:00}:
efmCuPmeErrorThreshMonEnable: -----> {false}:
efmCuPmeErrorThreshMonNotifyEnable: -> {false}:
efmCuPmeErrorThreshMonInterval: -----> {12}:
efmCuPmeErrorThreshMonClrInterval: --> {1800}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Here is the error message that is returned when the data rate is set too high for the constellation type.

```

Pme Data Rate and Constellation values are not compatible
tcpam32 has a Max Data Rate of 10176

```

Here is the error message that is returned when the data rate is set too low for the constellation type.

Pme Data Rate value invalid for this device

Here is the error message if you try to set an adaptive rate for TCPAM that does not support the adaptive rate:

```
zSH> update pme-profile 1/1/1/0/shdsl
pme-profile 1/1/1/0/shdsl
Please provide the following: [q]uit.
efmCuPmeAdminSubType: -----> {ieee2baset1r}:
efmCuPmeAdminProfile: -----> {0}:
efmCuPAFRemoteDiscoveryCode: -----> {}:
efmCuPmeThreshLineAtn: -----> {0}:
efmCuPmeThreshMinSnrMgn: -----> {0}:
efmCuPmeLineAtnCrossingEnable: -----> {false}:
efmCuPmeSnrMgnCrossingTrapEnable: ---> {false}:
efmCuPmeDeviceFaultEnable: -----> {false}:
efmCuPmeConfigInitFailEnable: -----> {false}:
efmCuPmeProtocolInitFailEnable: -----> {false}:
efmCuPme2BProfileDescr: -----> {}:
efmCuPme2BRegion: -----> {region1}:
efmCuPme2BDataRate: -----> {14912}: 0
efmCuPme2BPower: -----> {0}:
efmCuPme2BConstellation: -----> {tcpam4}: tcpam64
efmCuPme2BProfileRowStatus: -----> {active}:
efmCuPmeNtr: -----> {ntr-local-osc}:
efmCuPmeThreshMaxSnrMgnDelta: -----> {20}:
efmCuPmeMaintenanceMode: -----> {off}:
efmCuPmeMaintenanceStartTime: -----> {00:00}:
efmCuPmeMaintenanceEndTime: -----> {23:59}:
efmCuPmeSnrMonitoringInterval: -----> {1:00}:
efmCuPmeErrorThreshMonEnable: -----> {false}:
efmCuPmeErrorThreshMonNotifyEnable: -> {false}:
efmCuPmeErrorThreshMonInterval: -----> {12}:
efmCuPmeErrorThreshMonClrInterval: --> {1800}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Pme Data Rate and Constellation values are not compatible
tcpam64 has a Data Rate range of [768 - 12736]
Starting over....
efmCuPmeAdminSubType: -----> {ieee2baset1r}:
```

Set a region

For the **efmCuPme2BRegion** parameter, the regions are set as specified in the relevant Regional Annex of [G.991.2]. Regional settings place limitation on the maximum allowed data rate, power, and constellation. The possible values for this parameter are:

- Region 1
Annex A and F (North America)
- Region 2

Annex B and G (Europe)

You can only change regions when the link is down.

SNR monitoring for bonded G.SHDSL lines

This section describes how the MXK provides SNR monitoring for G.SHDSL lines:

- [SNR monitoring for the MXK, page 1430](#)
- [MXK SNR monitoring pme-profile parameters, page 1431](#)
- [Usage for SNR pme-profile and efm-port parameters, page 1433](#)
- [MXK SNR monitoring configuration, page 1434](#)
- [Verify SNR monitoring is enabled/disabled, page 1442](#)
- [G. SHDSL SNR monitoring example, page 1443](#)
- [Disable SNR monitoring, page 1448](#)

SNR monitoring for the MXK

This section describes how SNR monitoring for the MXK works:

- [SNR monitoring for the MXK overview, page 1430](#)
- [Current condition SNR maximum threshold, page 1431](#)
- [Current condition minimum SNR threshold, page 1431](#)

SNR monitoring for the MXK overview

Seamless Rate Adaptation (SRA) is found on ADSL2 lines to monitor the current Signal to Noise Ratio (SNR) and seamlessly adjusts the rate of the line to keep the current SNR at the target SNR. SRA prevents potential errors when the SNR drops by sacrificing train rate and allows for a higher train rate if the SNR improves. Because SHDSL lines do not have a comparable feature, when SHDSL forces a line to retrain to get a better rate, seamless retraining is not possible because traffic does not flow on links that are retraining.

For MXK G.SHDSL bonded lines, it is possible to configure SNR monitoring to monitor the current SNR and retrain the link when the SNR rises above or drops below target thresholds for current condition only. When retraining is performed on SHDSL bond groups of two or more ports, one link per bond group retrains at a time causing the performance of the bond group to become somewhat degraded but does not eliminate traffic flow entirely. Retraining on bond groups of just one port causes the bond group to go down. The system performs the retraining of G.SHDSL links only during a maintenance period that you must specify. This maintenance period can be an entire twenty four hour day or any portion of a twenty four hour day. After a link retrains, the MXK waits a maximum of 3 minutes for the link to come up before retraining the next port.

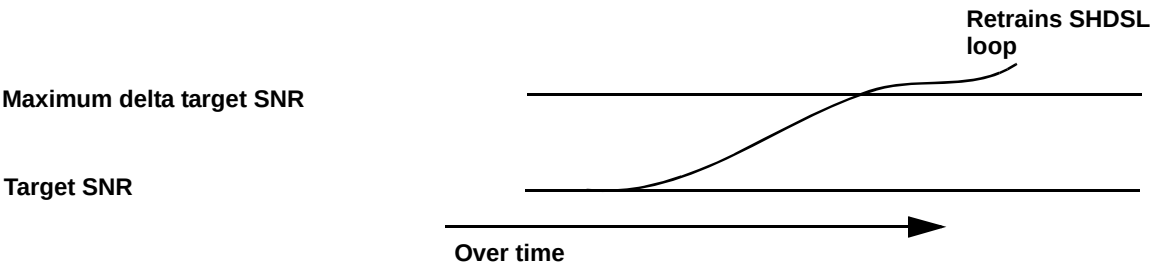
To configure SNR monitoring, a target SNR is set in the **efm-port** profile for current condition or worst case SNR. Additionally, a maximum delta SNR

threshold and a minimum SNR threshold is set in the **pme-profile**. SNR monitoring is configured for either current condition or for worst case SNR.

Current condition SNR maximum threshold

Current condition SNR determines the maximum threshold that causes the link to retrain by checking two SNR settings, the **efmCuTargetCurrentConditionSnrMgn** value in the **efm-port** profile and the **efmCuPmeThresMaxSnrMgnDelta** value in the **pme-profile**. When the current SNR exceeds the sum of these two values, the system retrains the link during the maintenance period.

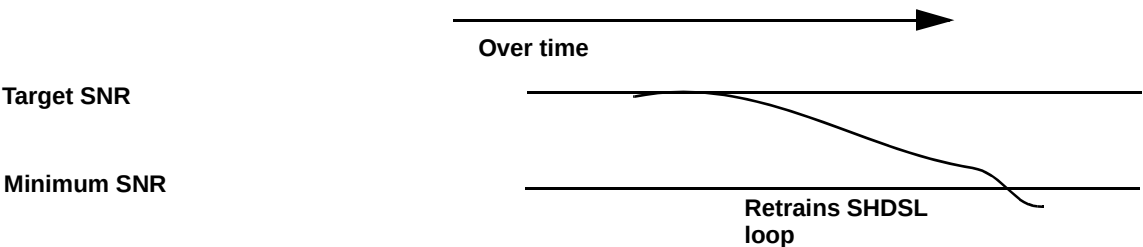
Figure 173: SNR monitoring for current condition maximum threshold



Current condition minimum SNR threshold

Current condition SNR determines the minimum threshold to cause the link to retrain by checking the **efmCuPmeThreshMinSnrMgn** value in the **pme-profile**. When the current SNR drops below this value, the system retrains the link during the maintenance period.

Figure 174: SNR monitoring for current condition minimum threshold



MXK SNR monitoring pme-profile parameters

The following parameters are used to configure the MXK for SNR monitoring and are set in the **pme-profile**:

- **efmCuPmeMaintenanceMode**

The variable settings of this parameter determine the current maintenance mode operational status. [Table 198](#) describes the **efmCuPmeMaintenanceMode** variables and their functions.

Table 198: SNR modes

Variable	Function
off (default)	The <i>off</i> setting indicates there is no SNR monitoring.
<i>manual</i>	The <i>manual</i> setting retrains the link one time only regardless of the SNR setting during the maintenance period set in the efmCuPmeMaintenanceStartTime and the efmCuPmeMaintenanceEndTime parameters.
<i>automatic-once</i>	The <i>automatic-once</i> setting retrains the link one time only during the maintenance period set in the efmCuPmeMaintenanceStartTime and the efmCuPmeMaintenanceEndTime parameters when the SNR value is outside of the specified threshold.
<i>automatic-daily</i>	The <i>automatic-daily</i> setting retrains the link each day during the maintenance period set in the efmCuPmeMaintenanceStartTime and the efmCuPmeMaintenanceEndTime parameters when the SNR value is outside of the specified threshold.
<i>automatic-continuous</i>	The <i>automatic-continuous</i> setting retrains the link according to the start and end times in the efmCuPmeMaintenanceStartTime and the efmCuPmeMaintenanceEndTime parameters at the interval set in the efmCuPmeSnrMonitoringInterval parameter when the SNR value is outside of the specified threshold.

- **efmCuPmeMaintenanceStartTime**

This parameter provides the start time for maintenance to retrain the link in *manual* maintenance mode. When maintenance mode is set to *automatic once* or *automatic daily*, this parameter provides the start time for monitoring of the SNR value that considers the SNR threshold values specified in **efmCuPmeThreshMinSnrMgn** and **efmCuPmeThreshMaxSnrMgnDelta** in the **pme-profile**.

The start time format is HH:MM where HH is the military time for hour (0-23) and MM is the military time for minutes (0-59).

- **efmCuPmeMaintenanceEndTime**

This parameter provides the end time for maintenance to retrain the link in *manual* maintenance mode. When maintenance mode is set to *automatic once* or *automatic daily*, this parameter provides the end time for monitoring of the SNR value that considers the SNR threshold values specified in **efmCuPmeThreshMinSnrMgn** and **efmCuPmeThreshMaxSnrMgnDelta** in the **pme-profile**.

The end time format is HH:MM where HH is the military time for hour (0-23) and MM is the military time for minutes (0-59).

- **efmCuPmeSnrMonitoringInterval**

This parameter provides the SNR monitoring interval for how frequently monitoring of the SNR occurs depending on the SNR threshold values specified in **efmCuPmeThreshMinSnrMgn** and **efmCuPmeThreshMaxSnrMgnDelta**.

The interval time format is HH:MM where HH is the military time for hour (0-23) and MM is the military time for minutes (0-59).

Usage for SNR pme-profile and efm-port parameters



Note: When an EFM link is configured for a fixed rate in the **pme-profile** member **efmCuPme2BDataRate** (a value other than 0) the SNR monitoring feature should not be used.

See [Configure automatic baud rate adaption and fixed rate settings on page 1424](#), [Verify SNR monitoring is enabled/disabled on page 1442](#) and [Disable SNR monitoring on page 1448](#).

For SNR monitoring, the maintenance mode settings **automatic-once**, **automatic-daily**, and **automatic-continuous** require the system to check for either the minimum or the maximum threshold settings. The minimum SNR threshold is set in the **efmCuPmeThreshMinSnrMgn** parameter in the **pme-profile**. The maximum SNR threshold is the sum of the **efmCuPmeThreshMaxSnrMgnDelta** in the **pme-profile** and the **efmCuTargetWorstCaseSnrMgn** or the **efmCuTargetCurrentConditionSnrMgn** in the **efm-port** profile. The target mode, either worst case or current condition, determines which variable settings are checked. These modes are set in the **efm-port** profile parameters **efmCuTargetCurrentConditionMode** or **efmCuTargetWorstCaseMode**.

- **efmCuPmeThreshMinSnrMgn**

For both current condition mode and worst case mode, forces the link to retrain to improve the SNR when the SNR falls below this setting. The default **efmCuPmeThreshMinSnrMgn** is 0 dB.

The link retrains during the maintenance window when the current **DslLineSnrMgn** is less than the **efmCuPmeThreshMinSnrMgn** setting.

Use the **dslstat** command to view the **DslLineSnrMgn** (in tenths dB).

$$(\text{DslLineSnrMgn}/10) < \text{efmCuPmeThreshMinSnrMgn}$$

- **efmCuPmeThreshMaxSnrMgnDelta**,
efmCuTargetWorstCaseSnrMgn, or
efmCuTargetCurrentConditionSnrMgn

For both worst case mode and current condition mode, forces the link to retrain to improve the SNR when the SNR rate rises above the sum of the **efmCuPmeThreshMaxSnrMgnDelta** and the **efmCuTargetWorstCaseSnrMgn** or the **efmCuTargetCurrentConditionSnrMgn**.

For worst case mode, use the **dslstat** command to view the **DslLineSnrMgn** (in tenths dB).

$(\text{DslLineSnrMgn}/10) > (\text{efmCuTargetWorstCaseSnrMgn} + \text{efmCuPmeThreshMaxSnrMgnDelta})$

For current condition mode, use the **dslstat** command to view the **DslLineSnrMgn** (in tenths dB).

$(\text{DslLineSnrMgn}/10) > (\text{efmCuTargetCurrentConditionSnrMgn} + \text{efmCuPmeThreshMaxSnrMgnDelta})$

MXK SNR monitoring configuration

This section provides MXK SNR configuration procedures:

- [Set SNR for target current condition or target worst case mode, page 1434](#)
- [Set MXK time and day, page 1435](#)
- [Set SNR monitoring from the CLI, page 1435](#)
- [View SNR monitoring statistics, page 1438](#)
- [Set SNR monitoring in the pme-profile, page 1439](#)
- [Configure SNR crossing traps, page 1442](#)



Note: If the maintenance mode, **minsnrnmgn**, **maxdeltasnrnmgn**, or the monitoring interval setting is changed, then the monitoring of the SNR values starts over again after 30 seconds.

Set SNR for target current condition or target worst case mode

SNR monitoring is set for either current condition mode or worst case mode. These modes are set to *true* or *false* in the **efm-port** profile parameters **efmCuTargetCurrentConditionMode** or **efmCuTargetWorstCaseMode**.

The target SNR value for current condition and worst case mode is set in the **efmCuTargetWorseCaseSnrMgn** (default 0db) and **efmCuTargetCurrentConditionSnrMgn** (default 6db) parameters.

Setting SNR for target current condition or target worst case mode

To set SNR monitoring to monitor for worst case SNR enter:

```
zSH> update efm-port 1-11-1-0/shdsl
efm-port 1-11-1-0/shdsl
Please provide the following: [q]uit.
efmCuPAFAdminState: -----> {enabled}:
efmCuPAFDiscoveryCode: -----> {}:
efmCuAdminProfile: -----> {0x01}:
```

```

efmCuTargetDataRate: -----> {50000}:
efmCuTargetWorstCaseSnrMgn: -----> {0}:
efmCuThreshLowBandwidth: -----> {0}:
efmCuLowBandwidthEnable: -----> {false}:
efmCuTargetCurrentConditionMode: ---> {false}: false
efmCuTargetCurrentConditionSnrMgn: -> {6}:
efmCuTargetWorstCaseMode: -----> {false}: true
efmCuPAFAutoDiscovery: -----> {disabled}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Set MXK time and day

Set the MXK time and day if not already set.

Setting the MXK time and day

You can either use the **setdatetime** command or use SNTP (the network time protocol) to provide time to the MXK rather than set it manually. For example:

```
zSH> setdatetime < month(mm) day(dd) year(yyyy) hour(hh) minute(mm) second(ss) >
```

Set SNR monitoring from the CLI

Use the **snrmon** command to enable SNR monitoring by setting minimum and maximum SNR margins for the link, and the administrative maintenance mode, intervals, and time for the link.

```
snrmon <modify|show|stats> <link interface>
```

Table 199: snrmon command

Variable	Function
link interface	Name and type of a physical interface. (i.e. 1-2-1-0/shdsl).
minsnr (minimum SNR margin)	Desired minimum snr margin threshold for the link.
deltasnr (maximum delta SNR margin)	Desired maximum snr margin threshold delta for the link.
mode	Desired administrative maintenance mode of the link. off manual automatic-once automatic-daily automatic-continuous

Table 199: snrmon command (Continued)

Variable	Function
trap	Determines if traps should be generated for threshold crossings of the link. enable disable
interval	Actual frequency that snr monitoring should occur for this link. HH:MM
start	Actual time of day (in 24 hour notation) that snr monitoring should start for this link. HH:MM
end	Actual time of day (in 24 hour notation) that snr monitoring should end for this link. HH:MM
Displayed statistics:	
Snr	Obtained SNR (in tenths DB) value for the physical line.
Snr Mgn Crossing Cnt	Count of each hour when SNR mgn is out of range.
Snr Min	Configured Min SNR value.
Snr Delta	Configured Delta SNR value.
Restart	Count of the number of times the port was restarted by snrmon feature.
Line Status	Current status of the port.

Setting SNR monitoring from the CLI

1 View the SHDSL SNR monitoring fields:

```
zSH> snrmon show 1-7-1-0/shdsl
                Shdsl Snr Monitoring
Port  Snr Min Snr Delta  Start    End    Interval  Trap    Mode
1      0      20      00:00   23:59   01:00   disabled  off
```

2 Set the desired minimum SNR margin for the link and verify the setting:

```
zSH> snrmon modify 1-7-1-0/shdsl minsnr 2

zSH> snrmon show 1-7-1-0/shdsl
                Shdsl Snr Monitoring
Port  Snr Min Snr Delta  Start    End    Interval  Trap    Mode
1      2      20      00:00   23:59   01:00   disabled  off
```

The **efmCuPmeThreshMinSnrMgn** parameter in the **pme-profile** is now 2.


```
efmCuPmeThreshMinSnrMgn: -----> {2}
```

3 Set the desired maximum SNR margin delta and verify the setting:

```
zSH> snrmon modify 1-7-1-0/shdsl deltasnr 16
```

```
zSH> snrmon show 1-7-1-0/shdsl
```

Shdsl Snr Monitoring									
Port	Snr	Min	Snr	Delta	Start	End	Interval	Trap	Mode
1	2		16		00:00	23:59	01:00	disabled	off

The **efmCuPmeThreshMaxSnrMgnDelta** parameter in the **pme-profile** is now set to 16.

```
efmCuPmeThreshMaxSnrMgnDelta: -----> {16}
```

4 Set the start of the maintenance period and verify the setting:

```
zSH> snrmon modify 1-7-1-0/shdsl start 01:00
```

```
zSH> s nrmonshow 1-7-1-0/shdsl
```

Shdsl Snr Monitoring									
Port	Snr	Min	Snr	Delta	Start	End	Interval	Trap	Mode
1	2		16		01:00	23:59	01:00	disabled	off

The **efmCuPmeMaintenanceStartTime** parameter in the **pme-profile** is now set to 1:00.

```
efmCuPmeMaintenanceStartTime: -----> {1:00}
```

5 Set the end of the maintenance period and verify the setting:

```
zSH> snrmon modify 1-7-1-0/shdsl end 02:00
```

```
zSH> snrmon show 1-7-1-0/shdsl
```

Shdsl Snr Monitoring									
Port	Snr	Min	Snr	Delta	Start	End	Interval	Trap	Mode
1	2		16		01:00	02:00	01:00	disabled	off

The **efmCuPmeMaintenanceEndTime** parameter in the **pme-profile** is now set to 2:00.

```
efmCuPmeMaintenanceEndTime: -----> {2:00}
```

6 Set the maintenance monitoring interval and verify the setting:

```
zSH> snrmon modify 1-7-1-0/shdsl interval 00:20
```

```
zSH> snrmon show 1-7-1-0/shdsl
```

Shdsl Snr Monitoring									
Port	Snr	Min	Snr	Delta	Start	End	Interval	Trap	Mode
1	2		16		01:00	02:00	00:20	disabled	off

```
1      2      16      01:00    02:00    00:20    disabled    off
```

The **efmCuPmeSnrMonitoringInterval** in the **pme-profile** is now set to **00:20**.

```
efmCuPmeSnrMonitoringInterval: -----> {00:20}
```

7 Enable the trap setting and verify the setting:

```
zSH> snrmon modify 1-7-1-0/shdsl trap enable
```

```
SH> snrmon show 1-7-1-0/shdsl
```

```

                Shdsl Snr Monitoring
Port  Snr Min Snr Delta  Start    End    Interval  Trap    Mode
1      2      16      01:00    02:00    00:20    enabled  off
```

The **efmCuPmeSnrMgnCrossingTrapEnable** parameter is now set to **true**:

```
efmCuPmeSnrMgnCrossingTrapEnable: ---> {true}
```

8 Set the SNR maintenance mode and verify the setting:

```
zSH> snrmon modify 1-7-1-0/shdsl mode automatic-daily
```

```
zSH> snrmon show 1-7-1-0/shdsl
```

```

                Shdsl Snr Monitoring
Port  Snr Min Snr Delta  Start    End    Interval  Trap    Mode
1      2      16      01:00    02:00    00:20    disabled  automatic-daily
```

The **efmCuPmeMaintenanceMode** parameter is now set to **automatic-daily**.

```
efmCuPmeMaintenanceMode: -----> {automatic-daily}
```

View SNR monitoring statistics

Viewing SNR monitoring statistics

Enter **snrmon stats** to view SNR error monitoring statistics:

```
zSH> snrmon stats 1-7-1-0/shdsl
```

```

                Shdsl Snr Monitoring Stats
                Snr          Snr
Port  (in tenths db) Crossing Cnt  Snr Min  Snr Delta  Restart  Line Status
1      180           0           0       20       0       ACT
```

Set SNR monitoring in the pme-profile

Zhone recommends using the CLI macro commands to configure SNR monitoring. Setting SNR monitoring in the **pme-profile** is for advanced system configuration users.

To set the maintenance mode for the MXK, enter the value for the **efmCuPmeMaintenanceMode** parameter of the **pme-profile**:

To set the maintenance period when SNR monitoring occurs, enter a start time and an end time for the **efmCuPmeMaintenanceStartTime** and the **efmCuPmeMaintenanceEndTime** parameters.

Setting the maintenance mode and time period in the pme-profile

- 1 View the current time on the MXK, if necessary, with **showdatetime**:

```
zSH> showdatetime
Current Time: TUE FEB 09 09:49:52 2010
```

- 2 For *automatic-once* mode, set the **efmCuPmeMaintenanceMode** to *automatic-once* and set the **efmCuPmeMaintenanceStartTime** and the **efmCuPmeMaintenanceEndTime** for the time of the maintenance period.

```
zSH> update pme-profile 1-5-1-0/shdsl
pme-profile 1-5-1-0/shdsl
Please provide the following: [q]uit.
efmCuPmeAdminSubType: -----> {ieee2baset1r}:
efmCuPmeAdminProfile: -----> {0}:
efmCuPAFRemoteDiscoveryCode: -----> {}:
efmCuPmeThreshLineAtn: -----> {0}:
efmCuPmeThreshMinSnrMgn: -----> {0}:
efmCuPmeLineAtnCrossingEnable: -----> {false}:
efmCuPmeSnrMgnCrossingTrapEnable: -> {false}:
efmCuPmeDeviceFaultEnable: -----> {false}:
efmCuPmeConfigInitFailEnable: -----> {false}:
efmCuPmeProtocolInitFailEnable: ---> {false}:
efmCuPme2BProfileDescr: -----> {}:
efmCuPme2BRegion: -----> {region1}:
efmCuPme2BDataRate: -----> {0}:
efmCuPme2BPower: -----> {0}:
efmCuPme2BConstellation: -----> {adaptive}:
efmCuPme2BProfileRowStatus: -----> {active}:
efmCuPmeNtr: -----> {ntr-local-osc}:
efmCuPmeThreshMaxSnrMgnDelta: -----> {20}:
efmCuPmeMaintenanceMode: -----> {off}: automatic-once
efmCuPmeMaintenanceStartTime: -----> {00:00}: 3:00
efmCuPmeMaintenanceEndTime: -----> {23:59}: 4:00
efmCuPmeSnrMonitoringInterval: -----> {01:00}:
efmCuPmeErrorThreshMonEnable: -----> {false}
efmCuPmeErrorThreshMonNotifyEnable: -> {false}
efmCuPmeErrorThreshMonInterval: -----> {12}
```

```
efmCuPmeErrorThreshMonClrInterval: --> {1800}
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

In this case, the SHDSL line will retrain one time only between 3:00 am and 4:00 am if the SNR value is outside of the specified threshold.

3 To verify the changes, enter:

```
zSH> get pme-profile 1-5-1-0/shdsl
pme-profile 1-5-1-0/shdsl
efmCuPmeAdminSubType: -----> {ieee2basetlr}
efmCuPmeAdminProfile: -----> {0}
efmCuPAFRemoteDiscoveryCode: -----> {}
efmCuPmeThreshLineAtn: -----> {0}
efmCuPmeThreshMinSnrMgn: -----> {0}
efmCuPmeLineAtnCrossingEnable: -----> {false}
efmCuPmeSnrMgnCrossingTrapEnable: ----> {false}
efmCuPmeDeviceFaultEnable: -----> {false}
efmCuPmeConfigInitFailEnable: -----> {false}
efmCuPmeProtocolInitFailEnable: -----> {false}
efmCuPme2BProfileDescr: -----> {}
efmCuPme2BRegion: -----> {region1}
efmCuPme2BDataRate: -----> {0}
efmCuPme2BPower: -----> {0}
efmCuPme2BConstellation: -----> {adaptive}
efmCuPme2BProfileRowStatus: -----> {active}
efmCuPmeNtr: -----> {ntr-local-osc}
efmCuPmeThreshMaxSnrMgnDelta: -----> {20}
efmCuPmeMaintenanceMode: -----> {automatic-once}
efmCuPmeMaintenanceStartTime: -----> {3:00}
efmCuPmeMaintenanceEndTime: -----> {4:00}
efmCuPmeSnrMonitoringInterval: -----> {01:00}
efmCuPmeErrorThreshMonEnable: -----> {false}
efmCuPmeErrorThreshMonNotifyEnable: -> {false}
efmCuPmeErrorThreshMonInterval: -----> {12}
efmCuPmeErrorThreshMonClrInterval: --> {1800}
```

4 For *automatic-daily* mode, set the **efmCuPmeMaintenanceMode** to *automatic-daily* and set the maintenance period start and end time in **efmCuPmeMaintenanceStartTime** and **efmCuPmeMaintenanceEndTime**.

```
zSH> update pme-profile 1-5-2-0/shdsl
pme-profile 1-5-2-0/shdsl
Please provide the following: [q]uit.
efmCuPmeAdminSubType: -----> {ieee2basetlr}:
efmCuPmeAdminProfile: -----> {0}:
efmCuPAFRemoteDiscoveryCode: -----> {}:
efmCuPmeThreshLineAtn: -----> {0}:
efmCuPmeThreshMinSnrMgn: -----> {0}:
efmCuPmeLineAtnCrossingEnable: -----> {false}:
efmCuPmeSnrMgnCrossingTrapEnable: ----> {false}:
```

```

efmCuPmeDeviceFaultEnable: -----> {false}:
efmCuPmeConfigInitFailEnable: -----> {false}:
efmCuPmeProtocolInitFailEnable: -----> {false}:
efmCuPme2BProfileDescr: -----> {}:
efmCuPme2BRegion: -----> {region1}:
efmCuPme2BDataRate: -----> {0}:
efmCuPme2BPower: -----> {0}:
efmCuPme2BConstellation: -----> {adaptive}:
efmCuPme2BProfileRowStatus: -----> {active}:
efmCuPmeNtr: -----> {ntr-local-osc}:
efmCuPmeThreshMaxSnrMgnDelta: -----> {20}:
efmCuPmeMaintenanceMode: -----> {off}: automatic-daily
efmCuPmeMaintenanceStartTime: -----> {00:00}: 1:00
efmCuPmeMaintenanceEndTime: -----> {23:59}: 2:00
efmCuPmeSnrMonitoringInterval: -----> {01:00}:
efmCuPmeErrorThreshMonEnable: -----> {false}:
efmCuPmeErrorThreshMonNotifyEnable: -> {false}:
efmCuPmeErrorThreshMonInterval: -----> {12}:
efmCuPmeErrorThreshMonClrInterval: --> {1800}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

In this case the SHDSL line will automatically retrain every day between 1:00 am and 2:00 am if the SNR value is outside of the specified threshold.

- 5 For the *automatic-continuous* mode, set the **efmCuPmeMaintenanceMode** to *automatic-continuous*, set the **efmCuPmeMaintenanceStartTime** and the **efmCuPmeMaintenanceEndTime** for the time of the maintenance period, and set the monitoring interval in the **efmCuPmeSnrMonitoringInterval**. The monitoring interval determines how often during the maintenance period the system checks to retrain the SNR rate.

```

zSH> update pme-profile 1-5-3-0/shdsl
pme-profile 1-5-3-0/shdsl
Please provide the following: [q]uit.
efmCuPmeAdminSubType: -----> {ieee2baset1r}:
efmCuPmeAdminProfile: -----> {0}:
efmCuPAFRemoteDiscoveryCode: -----> {}:
efmCuPmeThreshLineAtn: -----> {0}:
efmCuPmeThreshMinSnrMgn: -----> {0}:
efmCuPmeLineAtnCrossingEnable: -----> {false}:
efmCuPmeSnrMgnCrossingTrapEnable: ---> {false}:
efmCuPmeDeviceFaultEnable: -----> {false}:
efmCuPmeConfigInitFailEnable: -----> {false}:
efmCuPmeProtocolInitFailEnable: -----> {false}:
efmCuPme2BProfileDescr: -----> {}:
efmCuPme2BRegion: -----> {region1}:
efmCuPme2BDataRate: -----> {0}:
efmCuPme2BPower: -----> {0}:
efmCuPme2BConstellation: -----> {adaptive}:

```

```

efmCuPme2BProfileRowStatus: -----> {active}:
efmCuPmeNtr: -----> {ntr-local-osc}:
efmCuPmeThreshMaxSnrMgnDelta: -----> {20}:
efmCuPmeMaintenanceMode: -----> {off}: automatic-continuous
efmCuPmeMaintenanceStartTime: -----> {00:00}: 1:00
efmCuPmeMaintenanceEndTime: -----> {23:59}: 7:00
efmCuPmeSnrMonitoringInterval: -----> {01:00}: 00:30
efmCuPmeErrorThreshMonEnable: -----> {false}:
efmCuPmeErrorThreshMonNotifyEnable: -> {false}:
efmCuPmeErrorThreshMonInterval: -----> {12}:
efmCuPmeErrorThreshMonClrInterval: --> {1800}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

In this example the maintenance mode is set to **automatic-continuous**, the start and end time are set from 1:00 am to 7:00 am, and the monitoring interval is set to every half hour.

Configure SNR crossing traps

An SNR crossing trap is sent only when the SNR value has gone from within range of the threshold to exceeding the range or from exceeding the range of the threshold to within the range. SNR monitoring will occur at the frequency configured by the interval parameter (**efmCuPmeSnrMonitoringInterval**).

Configuring SNR crossing traps

Set **efmCuPmeSnrMgnCrossingTrapEnable** to *true*.

Verify SNR monitoring is enabled/disabled

You can verify that SNR monitoring is disabled either by using the **snrmon show** command for the interface or checking the parameter **efmCuPmeMaintenanceMode** in the **pme-profile** for the interface.

When you use the **snrmon show** command and the *Mode* displays **off**, SNR monitoring is disabled.

```
zSH> snrmon show 1-5-1-0/shdsl
```

Shdsl Snr Monitoring							
Port	Snr Min	Snr Delta	Start	End	Interval	Trap	
1	0	20	00:00	23:59	01:00	disabled	
off							

If the *Mode* shows any other value than off, SNR monitoring is enabled.

```
zSH> snrmon show 1-5-1-0/shdsl
```

Shdsl Snr Monitoring

Port	Snr Min	Snr Delta	Start	End	Interval	Trap
Mode						
1	0	20	00:00	23:59	01:00	disabled
automatic-daily						

G. SHDSL SNR monitoring example

A line which will retrain during the next maintenance interval, because the **DslLineSnrMgn** of 18 dB is greater than the sum of **efmCuTargetCurrentConditionSnrMgn** (6 dB) and **efmCuPmeThreshMaxSnrMgnDelta** (10 dB); all parameters involved in determining whether the retrain should occur are shown in bold.

In this example the maintenance period is for the entire day (from 00:00 to 23:59), so technically there is no next maintenance window since it is always within the maintenance window. The link will retrain every 10 minutes if **dslinesnrmgn** continues to exceed the set threshold since the interval parameter (**efmCuPmeSnrMonitoringInterval**) is 10 minutes.

```
zSH> get pme-profile 1-7-6-0/shdsl
pme-profile 1-7-6-0/shdsl
efmCuPmeAdminSubType: -----> {ieee2basetlr}
efmCuPmeAdminProfile: -----> {0}
efmCuPAFRemoteDiscoveryCode: -----> {}
efmCuPmeThreshLineAtn: -----> {0}
efmCuPmeThreshMinSnrMgn: -----> {0}
efmCuPmeLineAtnCrossingEnable: ----> {true}
efmCuPmeSnrMgnCrossingTrapEnable: -> {true}
efmCuPmeDeviceFaultEnable: -----> {false}
efmCuPmeConfigInitFailEnable: -----> {false}
efmCuPmeProtocolInitFailEnable: ---> {false}
efmCuPme2BProfileDescr: -----> {}
efmCuPme2BRegion: -----> {region1}
efmCuPme2BDataRate: -----> {0}
efmCuPme2BPower: -----> {0}
efmCuPme2BConstellation: -----> {adaptive}
efmCuPme2BProfileRowStatus: -----> {active}
efmCuPmeNtr: -----> {ntr-local-osc}
efmCuPmeThreshMaxSnrMgnDelta: -----> {10}
efmCuPmeMaintenanceMode: -----> {automatic-continuous}
efmCuPmeMaintenanceStartTime: -----> {00:00}
efmCuPmeMaintenanceEndTime: -----> {23:59}
efmCuPmeSnrMonitoringInterval: ----> {00:10}
```

```
zSH> get efm-port 1-7-6-0/shdsl
efmCuPAFAdminState: -----> {enabled}
efmCuPAFDiscoveryCode: -----> {}
efmCuAdminProfile: -----> {0x01}
efmCuTargetDataRate: -----> {50000}
efmCuTargetWorstCaseSnrMgn: -----> {0}
efmCuThreshLowBandwidth: -----> {0}
efmCuLowBandwidthEnable: -----> {false}
efmCuTargetCurrentConditionMode: ---> {true}
```

```
efmCuTargetCurrentConditionSnrMgn: -> {6}
efmCuTargetWorstCaseMode: -----> {false}
```

Use the **dslstat** command to verify the DSL Line Margin (DslLineSnrMgn) in bold which is displayed in tenths of a dB, so it is 18 dB

```
zSH> dslstat 1-7-6-0/shdsl
```

General Stats:

```
-----
AdminStatus.....UP
LineStatus.....DATA
Line uptime
(DD:HH:MM:SS).....0:00:00:47
DslUpLineRate (bitsPerSec).....5696000
DslDownLineRate (bitsPerSec).....5696000
DslMaxAttainableUpLineRate (bitsPerSec).....5696000
DslMaxAttainableDownLineRate (bitsPerSec)....5696000
Out Octets.....489212764
Out Pkts/Cells.....1913588
Out Discards.....0
Out Errors.....0
In Octets.....12034672
In Pkts/Cells.....161629
In Discards.....0
In Errors.....0
```

DSL Physical Stats:

```
-----
DslLineSnrMgn (tenths dB).....180
DslLineAtn (tenths dB).....0
DslCurrOutputPwr (tenths dB).....0
LOFS.....0
LOLS.....0
LOSS.....24
ESS.....1618
CRC Errors.....42
Inits.....139
```

A line which will NOT retrain because the DslLineSnrMgn of 2 dB is greater than the efmCuPmeThreshMinSnrMgn of 0 dB, and the DslLineSnrMgn of 2 dB is less than the sum of efmCuTargetWorstCaseSnrMgn (0 dB) and the efmCuPmeThreshMaxSnrMgnDelta (3 dB); all parameters involved in determining whether the retrain should occur are shown in bold.

```
zSHc> get pme-profile 1-7-4-0/shdsl
```

```
pme-profile 1-7-4-0/shdsl
efmCuPmeAdminSubType: -----> {ieee2baset1r}
efmCuPmeAdminProfile: -----> {0}
efmCuPAFRemoteDiscoveryCode: -----> {}
efmCuPmeThreshLineAtn: -----> {0}
efmCuPmeThreshMinSnrMgn: -----> {0}
efmCuPmeLineAtnCrossingEnable: ----> {false}
```



```

efmCuPmeSnrMgnCrossingTrapEnable: -> {true}
efmCuPmeDeviceFaultEnable: -----> {false}
efmCuPmeConfigInitFailEnable: -----> {false}
efmCuPmeProtocolInitFailEnable: ----> {false}
efmCuPme2BProfileDescr: -----> {}
efmCuPme2BRegion: -----> {region1}
efmCuPme2BDataRate: -----> {0}
efmCuPme2BPower: -----> {0}
efmCuPme2BConstellation: -----> {adaptive}
efmCuPme2BProfileRowStatus: -----> {active}
efmCuPmeNtr: -----> {ntr-local-osc}
efmCuPmeThreshMaxSnrMgnDelta: -----> {3}
efmCuPmeMaintenanceMode: -----> {automatic-continuous}
efmCuPmeMaintenanceStartTime: -----> {00:00}
efmCuPmeMaintenanceEndTime: -----> {23:59}
efmCuPmeSnrMonitoringInterval: ----> {01:00}

zSH> get efm-port 1-7-4-0/shdsl
efm-port 1-7-4-0/shdsl
efmCuPAFAdminState: -----> {enabled}
efmCuPAFDiscoveryCode: -----> {}
efmCuAdminProfile: -----> {0x01}
efmCuTargetDataRate: -----> {50000}
efmCuTargetWorstCaseSnrMgn: -----> {0}
efmCuThreshLowBandwidth: -----> {0}
efmCuLowBandwidthEnable: -----> {false}
efmCuTargetCurrentConditionMode: ----> {false}
efmCuTargetCurrentConditionSnrMgn: -> {6}
efmCuTargetWorstCaseMode: -----> {true}

```

Table 200 describes the error monitoring parameters in the **pme-profile**.

Table 200: Error monitoring parameters in the pme-profile

efmCuPmeErrorThreshMonEnable	Enables/disables the error threshold monitoring. Error threshold monitoring looks for errors on the physical line. When there have been efmCuPmeErrorThreshMonInterval number of consecutive seconds with errors, the line will first be taken down with a retrain in hopes of bettering the SNR to rectify the error situation. If, when the line comes up, we do not achieve efmCuPmeErrorThreshMonClrInterval number of consecutive error-free seconds before hitting another efmCuPmeErrorThreshMonInterval number of consecutive seconds with errors, the line will cease to be used for carrying traffic. The physical line will remain active so error monitoring can continue but data will no longer traverse until efmCuPmeErrorThreshMonClrInterval number of consecutive error-free seconds is achieved, at which time the line will resume carrying traffic.

Table 200: Error monitoring parameters in the pme-profile (Continued)

efmCuPmeErrorThreshMonNotifyEnable	Enables/disables the error threshold monitoring notification (via CLI or alarm manager) when an error threshold has been exceeded or cleared.
efmCuPmeErrorThreshMonInterval	Specifies the number of consecutive seconds of detecting errors that, once reached, will cause the physical line to be deemed a poor performer and causes an action to be taken. See efmCuPmeErrorThreshMonEnable .
efmCuPmeErrorThreshMonClrInterval	Specifies the number of consecutive error-free seconds that must be achieved in order to declare the physical line usable for carrying data traffic. See efmCuPmeErrorThreshMonEnable .

Compare the with the **DslLineSnrMgn** line statistics using the **dslstat** command:

```
zSH> dslstat 1-7-4-0/shdsl
```

General Stats:

```
AdminStatus.....UP
LineStatus.....DATA
Line uptime(DD:HH:MM:SS).....0:19:13:44
DslUpLineRate (bitsPerSec).....2576000
DslDownLineRate (bitsPerSec).....2576000
DslMaxAttainableUpLineRate (bitsPerSec).....5696000
DslMaxAttainableDownLineRate (bitsPerSec)....5696000
Out Octets.....9372
Out Pkts/Cells.....64
Out Discards.....0
Out Errors.....0
In Octets.....7961682
In Pkts/Cells.....241214
In Discards.....0
In Errors.....411809
```

DSL Physical Stats:

```
DslLineSnrMgn (tenths dB).....20
DslLineAtn (tenths dB).....210
DslCurrOutputPwr (tenths dB).....0
LOFS.....0
LOLS.....0
LOSS.....0
ESS.....229
CRC Errors.....0
Inits.....11
```

Table 201: SHDSL statistics

General statistics	Description
AdminStatus	Administrative status of the port: Values: Up Interface is ready to pass packets. Down Interface is unable to pass packets. Testing Interface is in a special testing state and is unable to pass packets.
LineStatus	Line status provides information about the SHDSL link. Values for a single SHDSL line: ACT — the line currently has link and can pass traffic in both directions OOS — the line does not have link TRAFFIC DISABLE — The line currently has link but not underlying SHDSL protocol; traffic will not pass.
Line uptime (DD:HH:MM:SS)	How long the interface has been up in dd hh mm (day, hour, minute, second) format.
DslUpLineRate (bitsPerSec)	Displays the DSL upstream (customer premise > central office) line rate on this interface.
DslDownLineRate (bitsPerSec)	Displays the DSL downstream (central office > customer premise) line rate on this interface.
DslMaxAttainableUpLineRate (bitsPerSec)	Displays the maximum line rate that can be supported on this line in the upstream direction.
DslMaxAttainableDownLineRate (bitsPerSec)	Displays the maximum line rate that can be supported on this line in the downstream direction.
Out Octets	Number of transmitted octets.
Out Pkts/Cells	Number of transmitted packets/cells
Out Discards	Number of transmission discards.
Out Errors	Number of transmission errors.
In Octets	Number of received octets.
In Pkts/Cells	Number of transmitted packets/cells
In Discards	Number of received discards.
In Errors	Number of receive errors.
SHDSL Physical Stats:	
DslLineSnrMgn (tenths dB)	DSL Line Signal to Noise Ratio Margin — The strength of the DSL signal relative to the noise on line.

Table 201: SHDSL statistics (Continued)

General statistics	Description
DslLineAtn (tenths dB)	DSL Line Attenuation — Measure of the signal degradation between the SHDSL port and the modem.
DslCurrOutputPwr (tenths dB)	Not currently used.
LOFS	Number of Loss of Frame Seconds.
LOLS	Number of Loss of Line Seconds.
LOSS	Number of Loss of Signal Seconds.
ESS	Number of errored seconds (the number of one-second intervals containing one or more CRC anomalies or one or more LoS or Sef defects) that has been reported in the current 15-minute interval.
CRC Errors	Cyclic Redundancy Check Errors — CRC Checks for transmission errors. The CRC code is computed from the data in the message. If the data is altered the CRC computation will not be in agreement with the data.
InitS	Number of line initialization attempts, including both successful and failed attempts.

Disable SNR monitoring

To disable SNR monitoring for a link use the **snrmon modify** command with **mode** set to **off**.

```
zSH> snrmon modify 1-5-1-0 mode off
```

SHDSL error monitoring

This section describes:

- [SHDSL error monitoring statistics, page 1449](#)
- [SHDSL error monitoring fields, page 1449](#)

SHDSL error monitoring statistics

The MXK provides the **errmon** command to view SHDSL error monitoring information.

Enter the **errmon stats interface/type** command to view SHDSL error monitoring statistics:

```
zSH> errmon stats 1-7-3-0/shdsl
          Shdsl Error Monitoring Stats
                                Max
Port  TC Down   CRC   ES   SES   Err Sec  Restart  Line Status
  3      0      16  32427  32416      0        0          ACT
```

[Table 202](#) defines the **errmon stats** fields.

Table 202: Definitions of displayed statistics

Parameter	Description
TC Down	Count of how many times the TC layer went down since the physical link was obtained.
CRC	Count of CRC anomalies.
ES	Count of one second intervals during which one or more CRCs are reported.
SES	Count of one second intervals during which at least 50 CRCs are reported.
Max Errored Sec	Maximum consecutive seconds with errors without causing action to be taken by errmon features.
Restart	Count of the number of times the port was restarted by errmon features.
Line Status	Current status of the port.

SHDSL error monitoring fields

The **errmon interface/type** command monitors the physical interface, for example, *1-5-1-0/shdsl*.

Enter **errmon show** to view the SHDSL error monitoring fields and enter **errmon modify** to enable the notify and monitor fields and to modify the default values.

```
zSH> errmon show 1-7-3-0/shdsl
```

```

                Shdsl Error Monitoring
Port  Monitor  Notify  ErrorInt  ClearInt  Line Status
3    disabled disabled    12      1800      ACT

```

The *Monitor* and the *Notify* fields must be enabled from the CLI:

Enable *Monitor*:

```
zSH> errmon modify 1-7-3-0/shdsl monitor enable
```

```

zSH> errmon show 1-7-3-0/shdsl
                Shdsl Error Monitoring
Port  Monitor  Notify  ErrorInt  ClearInt  Line Status
3    enabled disabled    12      1800      ACT

```

Enable *Notify*:

```
zSH> errmon modify 1-7-3-0/shdsl notify enable
```

```

zSH> errmon show 1-7-3-0/shdsl
                Shdsl Error Monitoring
Port  Monitor  Notify  ErrorInt  ClearInt  Line Status
3    enabled  enabled    12      1800      ACT

```

To change *errinterval* values enter:

```
zSH> errmon modify 1-7-3-0/shdsl errinterval 10
```

```

zSH> errmon show 1-7-3-0/shdsl
                Shdsl Error Monitoring
Port  Monitor  Notify  ErrorInt  ClearInt  Line Status
3    enabled  enabled    10      1800      ACT

```

To change *clrinterval* values enter:

```
zSH> errmon modify 1-7-3-0/shdsl clrinterval 600
```

```

zSH> errmon show 1-7-3-0/shdsl
                Shdsl Error Monitoring
Port  Monitor  Notify  ErrorInt  ClearInt  Line Status
3    enabled  enabled    10        600      ACT

```

Table 203: errmon parameters

Parameter	Description
link interface (port)	Name and type of a physical interface. For example, 1-5-1-0/shdsl.
monitor	Determines if error monitoring should be performed on the link.
notify	Determines if a notification to the CLI, alarm manager, and ZMS should be generated when an error threshold is exceeded or cleared on the link.

Table 203: errmon parameters (Continued)

Parameter	Description
errinterval	Specifies the number of consecutive seconds of detecting errors that, once reached, causes the physical line to be considered a poor performer and action to be taken.
clrinterval	Specifies the number of consecutive error-free seconds that must be achieved in order to declare the physical line usable for transporting data traffic.

errmon modify

Syntax `error modify|show|stats link/interface`

SHDSL statistics

Verifying the interface

Use the **dslstat** command to display the status of an SHDSL interface:

```
zSH> dslstat 1-7-4-0/shdsl
General Stats:
-----
AdminStatus.....UP
LineStatus.....DATA
Line uptime (DD:HH:MM:SS).....0:01:18:41
DslUpLineRate (bitsPerSec).....5696000
DslDownLineRate (bitsPerSec).....5696000
DslMaxAttainableUpLineRate (bitsPerSec).....5696000
DslMaxAttainableDownLineRate (bitsPerSec)....5696000
In Octets.....307992514
In Pkts/Cells/Frags.....1294816
In Discards.....0
In Errors.....0
Out Octets.....0
Out Pkts/Cells/Frags.....0
Out Discards.....0
Out Errors.....0
DSL Physical Stats:
-----
DslLineSnrMgn (tenths dB).....180
DslLineAtn (tenths dB).....0
DslCurrOutputPwr (tenths dB).....0
LOFS.....0
LOLS.....0
LOSS.....0
ESS.....57
CRC Errors.....3
Inits.....2
```

[Table 204](#) defines the statistics displayed in the **dslstat** command for an SHDSL line.

Clearing DSL counters (SHDSL)

You can clear DSL counters to make identifying the changing statistics easier to read.

- 1 Clear the statistics using the **dslstat clear** command

```
zSH> dslstat clear 1-7-4-0/shdsl
```

- 2 View the changes

For reference the `dslstat` command ([Use the `dslstat` command to display the status of an SHDSL interface: on page 1452](#)) shows the statistics prior to clearing the statistics. Statistics which are cleared by the **`dslstat clear`** command are in *bold*.

```
zSH> dslstat clear 1-7-4-0/shdsl
zSH> dslstat 1-7-4-0/shdsl

General Stats:
-----
AdminStatus.....UP
LineStatus.....DATA
Line uptime (DD:HH:MM:SS).....0:04:14:58
DslUpLineRate (bitsPerSec).....832000
DslDownLineRate (bitsPerSec).....832000
DslMaxAttainableUpLineRate (bitsPerSec).....5696000
DslMaxAttainableDownLineRate (bitsPerSec)....5696000
In Octets.....0
In Pkts/Cells/Frags.....0
In Discards.....0
In Errors.....0
Out Octets.....0
Out Pkts/Cells/Frags.....0
Out Discards.....0
Out Errors.....0

DSL Physical Stats:
-----
DslLineSnrmgn (tenths dB).....80
DslLineAtn (tenths dB).....280
DslCurrOutputPwr (tenths dB).....0
LOFS.....0
LOLS.....0
LOSS.....0
ESS.....0
CRC Errors.....0
Inits.....0
```

Table 204: SHDSL statistics

General statistics	Description
AdminStatus	Administrative status of the port: Values: Up Interface is ready to pass packets. Down Interface is unable to pass packets. Testing Interface is in a special testing state and is unable to pass packets.

Table 204: SHDSL statistics (Continued)

General statistics	Description
LineStatus	Line status provides information about the SHDSL link. For information about the status on EFM or N2N bond groups, see Bond group statistics and port statistics on page 1456. Values for a single SHDSL line: ACT — the line currently has link and can pass traffic in both directions OOS — the line does not have link TRAFFIC DISABLE — The line currently has link but not underlying SHDSL protocol; traffic will not pass.
Line uptime (DD:HH:MM:SS)	How long the interface has been up in dd hh mm (day, hour, minute, second) format.
DslUpLineRate (bitsPerSec)	Displays the DSL upstream (customer premise > central office) line rate on this interface.
DslDownLineRate (bitsPerSec)	Displays the DSL downstream (central office > customer premise) line rate on this interface.
DslMaxAttainableUpLineRate (bitsPerSec)	Displays the maximum line rate that can be supported on this line in the upstream direction.
DslMaxAttainableDownLineRate (bitsPerSec)	Displays the maximum line rate that can be supported on this line in the downstream direction.
Out Octets	Number of transmitted octets.
Out Pkts/Cells/Frags	Number of transmitted packets/cells/fragments
Out Discards	Number of transmission discards.
Out Errors	Number of transmission errors.
In Octets	Number of received octets.
In Pkts/Cells/Frags	Number of transmitted packets/cells/fragments
In Discards	Number of received discards.
In Errors	Number of receive errors.
SHDSL Physical Stats:	
DslLineSnrMgn (tenths dB)	DSL Line Signal to Noise Ratio Margin — The strength of the DSL signal relative to the noise on line.
DslLineAtn (tenths dB)	DSL Line Attenuation — Measure of the signal degradation between the SHDSL port and the modem.
DslCurrOutputPwr (tenths dB)	Not currently used.
LOFS	Number of Loss of Frame Seconds.
LOLS	Number of Loss of Line Seconds.

Table 204: SHDSL statistics (Continued)

General statistics	Description
LOSS	Number of Loss of Signal Seconds.
ESS	Number of errored seconds (the number of one-second intervals containing one or more CRC anomalies or one or more LoS or Sef defects) that has been reported in the current 15-minute interval.
CRC Errors	Cyclic Redundancy Check Errors — CRC Checks for transmission errors. The CRC code is computed from the data in the message. If the data is altered the CRC computation will not be in agreement with the data.
Inits	Number of line initialization attempts, including both successful and failed attempts.

Bond group statistics and port statistics

The MXK provides the ability to view statistics for both each port associated with a link in a bond group and an individual bond group.

- [View port statistics, page 1456](#)
- [View bond group statistics, page 1457](#)

View port statistics

EFM bonding fragments packets across multiple lines so that packet counts for each EFM port indicates the number of EFM packet fragments for that port. At the physical port level, unicast packet counts show the number of packet fragments for that port. Octets for the physical port include all bytes received, including those from errored packet fragments and protocol overhead.

Data in the **dslstat** command is provided for each port associated with a link in the bond group. The data is collected differently for N2N and EFM ports and bond groups.

View each of the 24 SHDLS ports by entering **dslstat shelf-slot-port-subport/type**:

```
zSH> dslstat 1-5-1-0/shdsl
General Stats:
-----
AdminStatus.....UP
LineStatus.....DATA
Line uptime (DD:HH:MM:SS).....0:04:56:27
DslUpLineRate (bitsPerSec).....5696000
DslDownLineRate (bitsPerSec).....5696000
DslMaxAttainableUpLineRate (bitsPerSec).....5696000
DslMaxAttainableDownLineRate (bitsPerSec)....5696000
Out Octets.....0
Out Pkts/Cells.....0
Out Discards.....0
Out Errors.....0
In Octets.....163790
In Pkts/Cells.....904
In Discards.....0
In Errors.....0
DSL Physical Stats:
-----
DslLineSnrMgn (tenths dB).....190
DslLineAtn (tenths dB).....10
DslCurrOutputPwr (tenths dB).....0
LOFS.....0
LOLS.....0
LOSS.....3
ESS.....149
CRC Errors.....0
Inits.....3
```

View bond group statistics

The MXK and other bonding capable devices provide the **bond stats** *shelf-slot-port-subport/type* command to display both the status of the bond group and the status of each individual link in the bond group and to provide statistics for the bond group. A bond group is the aggregate of individual links on a device connected to the same CPE that provides a higher bandwidth than individual links can provide.

To view the statistics for an MXK bond group enter **bond stats** *shelf-slot-port-subport/type*:

```
zSH> bond stats 1-6-201-0/efmbond
***** Bond group statistics *****
                        Group Info
Slot      GrpId      Interface Name
   6       185      1-6-201-0/efmbond
AdminStatus  OperStatus  Bandwidth      Last Change
   UP         UP         45568000       0.04:14:53
Threshold Alarm Config
disabled

                        Group Members
Port      Admin  Oper   Bandwidth (tenths dB)  Snr      Tc Layer  Interface Name
6/2      UP      UP     5696000      180       0      1-6-2-0/shdsl
6/1      UP      UP     5696000      180       0      1-6-1-0/shdsl
6/8      UP      UP     5696000      180       0      1-6-8-0/shdsl
6/7      UP      UP     5696000      170       0      1-6-7-0/shdsl
6/6      UP      UP     5696000      170       0      1-6-6-0/shdsl
6/5      UP      UP     5696000      180       0      1-6-5-0/shdsl
6/4      UP      UP     5696000      180       0      1-6-4-0/shdsl
6/3      UP      UP     5696000      180       0      1-6-3-0/shdsl

Statistics (Received)
Octets      108011
Ucast       5
Mcast       737
Bcast       10
Discards    0
Errors      0

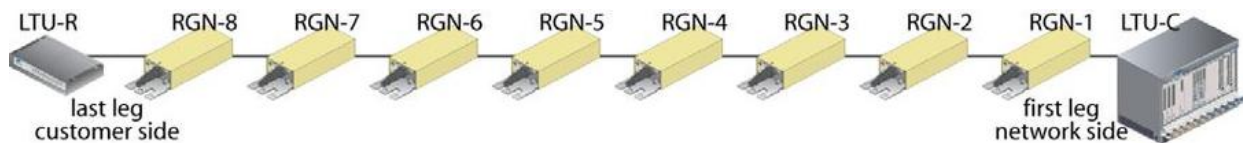
Statistics (Transmitted)
Octets      2390
Ucast       10
Mcast       0
Bcast       5
Discards    0
```

EtherXtender statistics

In order to improve troubleshooting of SHDSL EtherXtender repeater hops, the ability to review statistics and SNR data on either/both of the CO and CPE ends of the SHDSL circuit has been added.

When the EtherXtender SHDSL line extender is used on the MXK, the **regenstats** command outputs SHDSL EtherXtender regenerator performance statistics as gathered by each EtherXtender on the line. The statistics are shown per line regardless of whether multiple lines are bonded.

Figure 175: In the display each EtherXtender is referred to as a regenerator (RGN)



The **regenstats** command displays the SHDSL statistics or clears SHDSL indexes:

```
regenstats <show|clear> <link interface>
```

The **regenstats show** command displays a number of SHDSL statistics. A list of the statistics and definitions are shown in [EtherXtender regenerator performance statistics, page 1461](#).

The **regenstats clear** command clears the SHDSL indexes on the specified link interface:

- SHDSL count of CRC Anomalies
- SHDSL count of Errored Seconds
- SHDSL count of Severely Errored Seconds
- SHDSL count of Unavailable Seconds
- SHDSL Loss of Sync Word Second (LOSWS) defect count

To display EtherXtender statistics

The CLI **regenstats show** command displays eight EtherXtender columns regardless of the number of EtherXtenders which are connected on the port.

LTU-C and LTU-R represent the Line Termination Units at the CO and CPE ends of the line, respectively. RGN designates the EtherXtender regenerators.

Notice that there is a reported value for the network side (e.g., SNR NET) and the customer side (e.g., SNR CST) of each EtherXtender. However, there is only one side for the LTU-C (customer side only) and LTU-R (network side only).

When the SNR and the loop attenuation display -1 (or 65535) it means that there is no EtherXtender for that number. NA is used to designate items in the

table which do not exist, like the network facing side from the CO and customer facing side from the CPE. The EtherXtenders will have both network and customer facing units.

1 Display the statistics for a complete set of eight EtherXtenders

```
zSH> regenstats show 1-9-1-0/shdsl
```

SHDSL EtherXtender Regenerator Performance Statistics

```
-----
SLOT      9
PORT      1
STATUS    UP
US RATE   5696000
DS RATE   5696000
-----
```

```
-----
                LTU-C  RGN-1  RGN-2  RGN-3  RGN-4  RGN-5
RGN-6  RGN-7  RGN-8  LTU-R
-----
```

```
-----
SNR NET      NA    200    200    200    200    200
200    200    200    190
SNR CST      160    200    200    190    200    200
200    200    190    NA
LOOPATN N     NA     0    10    10     0    10
0    10    10     0
LOOPATN C     10    10     0     0     0    10
10     0     0    NA
CRC NET      NA     0     0     0     0     0
0     0     0     2
CRC CST       3     4     0     3     1     0
0     5     0    NA
ES NET      NA     0     0     0     0     0
0     0     0    73
ES CST      18    10     7     7     6     5
3     3     1    NA
SES NET      NA     0     0     0     0     0
0     0     0    73
SES CST      17     8     7     6     5     5
3     1     1    NA
UAS NET      NA     0     0     0     0     0
0     0     0     0
UAS CST       0     0     0     0     0     0
0     0     0    NA
LOSWS NET     NA     0     0     0     0     0
0     0     0     1
LOSWS CST     36     8     7     6     5     5
3     1     1    NA
DC-CONT N     NA    NO     NO     NO     NO     NO
NO     NO     NO    NO
DC-CONT C     NO    NO     NO     NO     NO     NO
NO     NO     NO    NA
-----
```

```

-----
LTU-C = CO endpoint, LTU-R = CPE endpoint, RGN-X =
Regenerator
Network side = NET or N Customer Side = CST or C
-----

```



Note: While the EtherXtenders establish links, negotiate rates, and begin to report statistics the SNR NET rate of the far end device will show -1 (or 65535). This value will be displayed until the line is completely negotiated.

Wait for the SNR CST of the LTU-R column to display a proper value before basing analysis on the displayed statistics.

2 Display the statistics for a set of four EtherXtenders

```
zSH> regenstats show 1-9-1-0/shdsl
```

```
-----
SHDSL EtherXtender Regenerator Performance Statistics
-----
```

```

SLOT      9
PORT      1
STATUS    UP
US RATE   5696000
DS RATE   5696000
-----

```

```

-----
                LTU-C  RGN-1  RGN-2  RGN-3  RGN-4  RGN-5
RGN-6  RGN-7  RGN-8  LTU-R
-----

```

```

-----
SNR NET      NA    190    200    200    200    -1
-1          -1    -1    180
SNR CST      170   200    200    200    190    -1
-1          -1    -1     NA
LOOPATN N    NA     0     10     0     0     -1
-1          -1    -1     0
LOOPATN C     0     0     0    10     0     -1
-1          -1    -1     NA
CRC NET      NA     0     0     0     0     0
0           0     0     5
CRC CST      0     0     3     0     0     0
0           0     0     NA
ES NET      NA     0     0     0     0     0
0           0     0    72
ES CST      8     4     4     2     1     0
0           0     0     NA
SES NET      NA     0     0     0     0     0
0           0     0    72
-----

```


SES CST	8	4	3	2	1	0
0 0	0	NA				
UAS NET	NA	0	0	0	0	0
0 0	0	0				
UAS CST	0	0	0	0	0	0
0 0	0	NA				
LOSWS NET	NA	0	0	0	0	0
0 0	0	19				
LOSWS CST	20	4	3	2	1	0
0 0	0	NA				
DC-CONT N	NA	NO	NO	NO	NO	NO
NO NO	NO	NO				
DC-CONT C	NO	NO	NO	NO	NO	NO
NO NO	NO	NA				

LTU-C = CO endpoint, LTU-R = CPE endpoint, RGN-X = Regenerator

Network side = NET or N Customer Side = CST or C

Table 205: EtherXtender regenerator performance statistics

UI label	Definition
SNR	The current SHDSL Signal-to-Noise Ratio (SNR) margin with respect to the received signal as perceived by the specific SHDSL modem in the span. The value is reported in tenths of a dB. A value of -1 (or 65535) is returned when the modem is down or not present.
Loop Attenuation	The current SHDSL Line Attenuation as perceived by the specific SHDSL modem in the span. The value is reported in tenths of a dB. A value of -1 (or 65535) is returned when the modem is down or not present.
CRC count	The current SHDSL count of CRC Anomalies as perceived by the specific SHDSL modem in the span. This value is reset when the port is down.
ES count	The current SHDSL count of Errored Seconds as perceived by the specific SHDSL modem in the span. This value is reset when the port is down.
SES count	The current SHDSL count of Severely Errored Seconds as perceived by the specific SHDSL modem in the span. This value is reset when the port is down.
UAS count	The current SHDSL count of Unavailable Seconds as perceived by the specific SHDSL modem in the span. This value is reset when the port is down.

Table 205: EtherXtender regenerator performance statistics

UI label	Definition
LOSWS count	The current SHDSL Loss of Sync Word Second (LOSWS) defect as perceived by the specific SHDSL modem in the span. This value is reset when the port is down.
DC Continuity Fault	The DC Continuity Fault indicator. This is used to indicate conditions that interfere with span powering such as shorts and open circuits. A value of No is returned when the port is down or initializing.

NOTE:

The LTU-C (CO) Node only has a Customer side modem.

The LTU-R (CPE) Node only has a Network side modem.

Regenerators 1-8 have both Network and Customer side modems.

To clear EtherXtender regenerator statistics

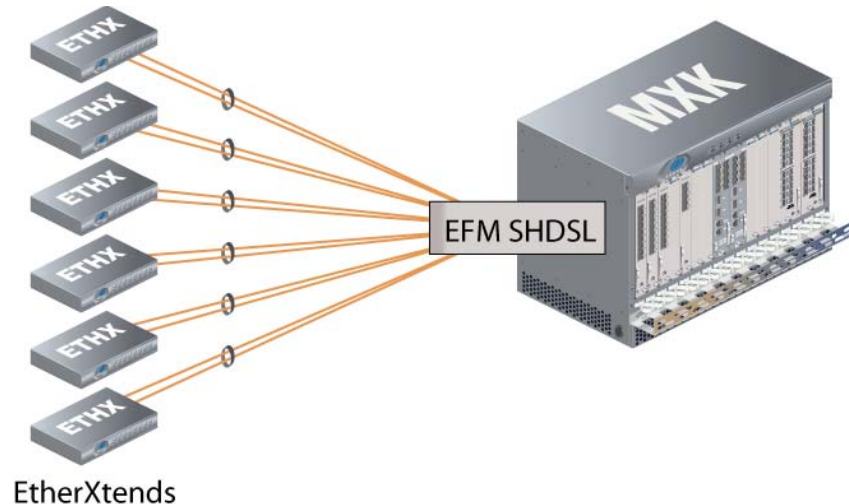
```
zSH> regenstats clear 1-9-1-0/shdsl
Clearing SHDSL Regenerator Stat Counts for SLOT 9 PORT
1
```

802.3ah EFM OAM

EFM OAM uses an in-band link layer OAM packet exchange between MXK EFM interfaces and OAM-capable CPEs, such as EtherXtend and EtherXtend Lite. The OAM-capable CPE functions as a remote peer to provide event notification. The EFM and N2N bond groups are Ethernet-like interfaces and support EFM OAM.

When EFM OAM is configured on a MXK EFM or Ethernet-like interface in active mode, the discovery process is started. If the interface peer also has OAM enabled, the discovery process continues until the peer is located. If the discovery process does not find a peer, the active interface continues sending the initial Information OAMPDU once a second until a peer OAM-enabled CPE device responds.

EFM OAM supports the MXK-EFM-SHDSL-24, MXK-EFM-SHDSL-24NTWC, MXK-EFM-SHDSL-24NTP card interfaces connected to EtherXtend and other compatible CPEs.



Configuring OAM support

The OAM interface is defined by an **ether-oam** profile that specifies the options for active/passive mode, loopback, and notification for events. By default, OAM is disabled on all MXK uplink and EFM interfaces.

To configure OAM features:

- 1 Create a new OAM profile for the desired EFM interface. By default, this profile is in passive mode with loopback disabled.

This example configures EFM OAM in active mode on EFM bond group 1-4-50-0/efmbond on a EFM-SHDSL-24 card in slot 4.

```
zSH> eth-oam add 1-4-50-0/efmbond active
```

- 2 Create a new OAM profile for the desired EtherXtend interface. By default, this profile is in passive mode with loopback disabled.

This example configures EFM OAM in passive mode on EFM bond group 1-1-40-0/efmbond on the peer EtherXtend.

```
zSH> eth-oam add 1-1-40-0/efmbond passive
```

3 Enter commands to modify and display OAM parameters.

The **eth-oam modify** command provides access to configurable settings in the ether-oam profile.

The **eth-oam show** command displays configured OAM settings.

The **eth-oam stats** command displays OAM statistics for a specified physical interface or bond group or all OAM interfaces.

eth-oam add

Configures and enables OAM interface on a physical interface.

Syntax `eth-oam interface/type [active | passive]`

Options `interface/type`

Name and type of the physical interface or bond group.

active

Sets OAM to active mode on this interface. The default is passive.

passive

Sets OAM to passive mode on this interface. The default is passive.

eth-oam delete

Deletes and disables the OAM configuration on the specified physical interface. This command does not delete any other configurations on this interface such as bond groups and bridge interfaces.

Syntax `eth-oam delete interface/type`

Options `interface/type`

Name and type of the physical interface or bond group.

eth-oam modify

Modifies a configured eth-oam interface.

Syntax `eth-oam modify interface/type [active | passive]`

Options `interface/type`

Name and type of the physical interface or bond group.

eth-oam show

Displays configured OAM parameters for the specified interface. If no interface is specified, configured OAM parameters are displayed for all OAM enabled interfaces.

Syntax `eth-oam show interface/type [peer]`

Options `interface/type`

Name and type of the physical interface or bond group.

peer

Displays the learned configuration information of the peer for the given interface. Includes peer MAC address, peer vendor OUI, peer vendor unique info, peer mode, peer max OAM PDU size, peer configuration revision, peer supported functions.

eth-oam stats

Displays OAM statistics for the specified interface. If no option is specified, statistics are displayed for all OAM interfaces.

Syntax `eth-oam stats interface/type`

Options `interface/type`

Name and type of the physical interface or bond group.

MXK-EFM-SHDSL-24 pinouts

The MXK-EFM-SHDSL-24 cards use standard RJ-21X pinouts. [Table 206](#) lists the port pinouts.

Table 206: MXK-EFM-SHDSL-24 Pinouts

Port	Pin Ring	Pin Tip
1	1	26
2	2	27
3	3	28
4	4	29
5	5	30
6	6	31
7	7	32
8	8	33
9	9	34

Table 206: MXK-EFM-SHDSL-24 Pinouts (Continued)

Port	Pin Ring	Pin Tip
10	10	35
11	11	36
12	12	37
13	13	38
14	14	39
15	15	40
16	16	41
17	17	42
18	18	43
19	19	44
20	20	45
21	21	46
22	22	47
23	23	48
24	24	49



Note: Pins 25 and 50 are not used.

Power and data connections for SHDSL CPE devices

This section describes the power connections on the MXK-EFM-SHDSL-24-NTP card that enable power to be delivered to Zhone SHDSL CPE devices and includes:

- [Deliver power and data to the CPE, page 1467](#)
- [Enable power on the SHDSL line, page 1469](#)

Deliver power and data to the CPE

The MXK-EFM-SHDSL-24-NTP card delivers power and data on the same wires. To deliver power to the CPE, two pairs of wires (four wires total) are required.

The specifications for the cables delivering power are as follows:

- 2 wires per port
- 26 AWG (0.4 mm) or 24 AWG (0.5 mm)

The LP IN port on the MXK-EFM-SHDSL-24-NTP card provides 12 pairs of wires to deliver power. The power is combined with the data and sent out over the 24 SHDSL ports to downstream CPE devices. One MXK-EFM-SHDSL-24-NTP card can provide power and data for up to 12 SHDSL CPE devices.

[Figure 176](#) illustrates the wiring connections for power and data being transmitted over the same pair of wires to a single CPE port. To power multiple CPE devices, use the pinouts described in [Table 206](#) to match SHDSL ports to the power pairs. Each set of four pins can power a single SHDSL CPE.

Figure 176: Example power and data delivered over the same wire pairs

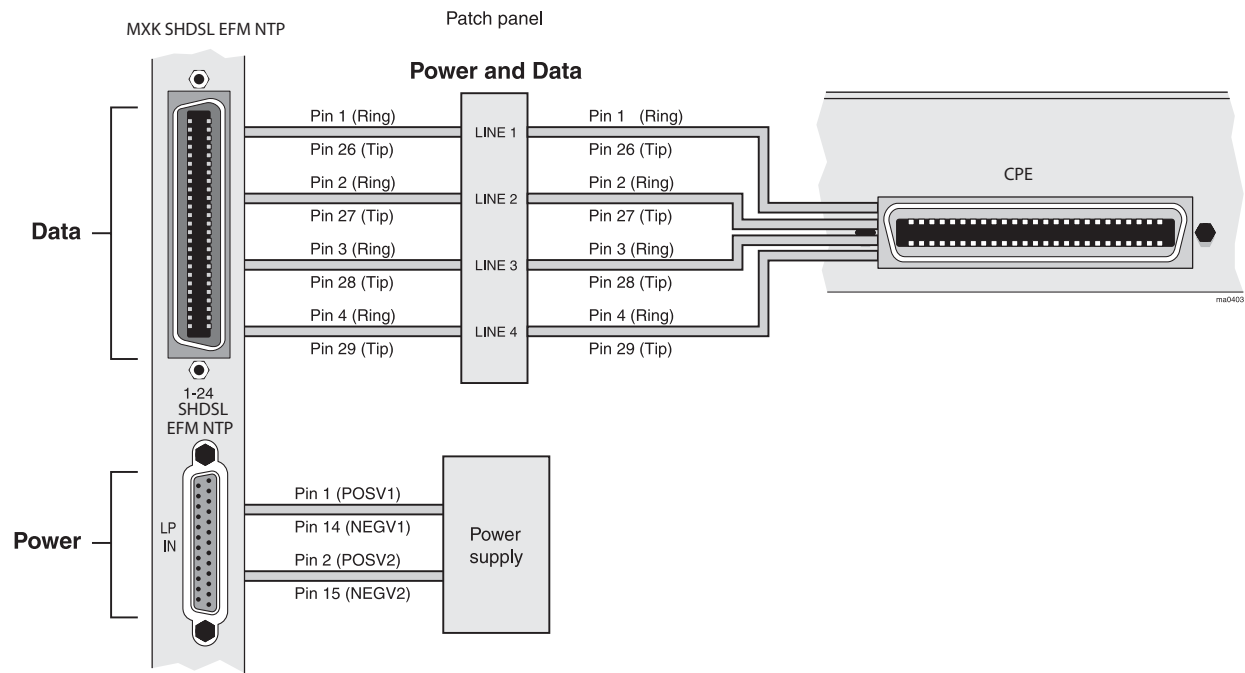


Table 207 describes the power connections for the 26 pin connector.

Table 207: Power connections between the line power pins and the SHDSL ports

Line power pin	SHDSL
pin 1	port 1
pin 14	port 2
pin 2	port 3
pin 15	port 4
pin 3	port 5
pin 16	port 6
pin 4	port 7
pin 17	port 8
pin 5	port 9
pin 18	port 10
pin 6	port 11
pin 19	port 12
pin 7	port 13
pin 20	port 14

Table 207: Power connections between the line power pins and the SHDSL ports

Line power pin	SHDSL
pin 8	port 15
pin 21	port 16
pin 9	port 17
pin 22	port 18
pin 10	port 19
pin 23	port 20
pin 11	port 21
pin 24	port 22
pin 12	port 23
pin 25	port 24

Enable power on the SHDSL line

The line power feature is enabled by default. This means that voltage is applied to the SHDSL line by default. This voltage comes from an external power supply as shown in [Figure 176](#). If the external power supply is not connected or turned off, voltage will simply not be supplied to the SHDSL line. However, the data stream will continue to be sent.

If someone needs to work on the line, voltage is removed from that line by setting adminstatus to maintenance. Maintenance mode stops the data stream and the voltage.



Note: When a port is set to maintenance mode, any MTAC testing that may be running on any port is turned off. Maintenance mode always has top priority.

```

zSH> update if-translate 1-13-1-0/shdsl
if-translate 1-13-1-0/shdsl
ifIndex: -----> {1421}
shelf: -----> {1}
slot: -----> {13}
port: -----> {1}
subport: -----> {0}
type: -----> {shdsl}
adminstatus: -----> {up} maintenance
physical-flag: -----> {true}
iftype-extension: --> {none}
ifName: -----> {1-13-1-0}
redundancy-param1: -> {0}

```



Note: The SHDSL line power feature requires that two lines are used together and both must be set to up in the adminstatus field. The lines do not need to be adjacent.

MTAC testing

The line power feature on the MXK-EFM-SHDSL-24-NTP card is mutually exclusive with MTAC testing and takes precedence over MTAC. When the line power feature is being used, MTAC testing cannot occur. To run MTAC testing, no ports on the MXK-EFM-SHDSL-24-NTP card can be in maintenance mode.

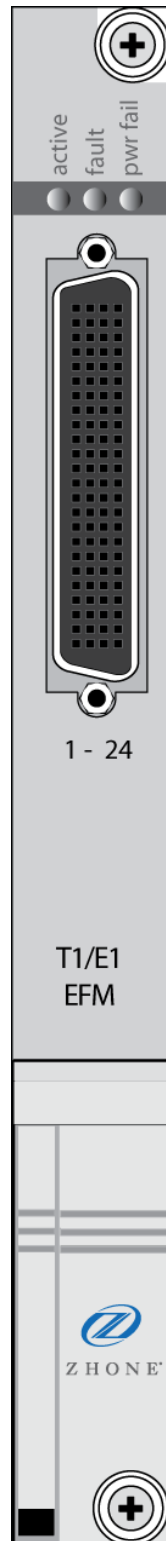
16

MXK EFM T1/E1 CARD

This chapter describes the MXK-EFM-T1/E1-24 card and explains how to configure it. It includes:

- [EFM T1/E1 card overview, page 1472](#)
- [EFM T1/E1 card specifications, page 1473](#)
- [EFM T1/E1 card configuration, page 1474](#)
- [Net-to-net bonding, page 1484](#)
- [Bond group statistics and port statistics, page 1488](#)
- [EFM T1/E1 24-port cables, page 1493](#)
- [Tests on the EFM T1/E1 card, page 1502](#)

EFM T1/E1 card overview



The MXK-EFM-T1/E1-24 card provides 24 T1/E1 bondable ports. The card provides Ethernet over T1 links to Zhone CPE devices. The T1 links can be added or removed as you configure your network. The card automatically performs load balancing over the link. The T1 links can be over dry copper four-wire pair or through a SONET fiber network that connects up to a T1 link on the far end. Both implementations transmit and receive over a Ds1 connection.

The MXK-EFM-T1/E1-24 bonded card supports bridging, VLANs, and Q-in-Q.

EFM T1/E1 cards on the MXK support up to 24 bond groups. Each bond group can have a maximum of eight members.

EFM T1/E1 card specifications

Table 208: MXK-EFM-T1/E1-24 card specifications

Specification	Description
Density	24 ports
Physical interface	Custom 96-pin Molex connector A cable is provided that breaks out to 4 non-terminated wire bundles for connecting to patch panels.
Size	1 slot
Cable	200-01365-01 Break out cable for use with patch panel. One 96-pin Molex connector to four 50-pin Champ telco connectors.
Line types	Frame formats supported: • D4 • ESF • E1 • E1-CRC Line codings: • AMI • B8ZS • HDB3
Supported line rates	1.544 MHz, 2.048 MHz
Redundancy	None
Power consumption	22 watts

EFM T1/E1 card configuration

This section describes how to configure the EFM T1/E1-24 card.

- [Create a card-profile for the EFM T1/E1 card, page 1474](#)
- [Activate a Ds1 interface, page 1477](#)
- [View the Ds1 interface, page 1477](#)

Create a card-profile for the EFM T1/E1 card

Each card installed in the system must have a **card-profile**. The type of line card determines the parameter settings in the **card-profile** and the software image for the card. Performing a **card add** automatically creates the **card-profile** for the card with the correct software image and settings.

[Table 209](#) describes the card type and software images for the MXK-EFM-T1E1-24 cards on the MXK:

Table 209: Card type and software image

Card	Type	Name of software image
MXK-EFM-T1E1-24	10214	<i>mxlc24t1e1bond.bin</i>

Creating the card profile for EFM T1E1-24 cards

Add a EFM T1E1-24 card to the system.

- 1 Install the EFM T1E1-24 card in the desired line card slot.
- 2 Create a **card-profile** for the card with the **card add slot linetype type** command.

When adding a EFM T1E1 card, the **card-profile** must be configured with either **linetype ds1** for T1 or **e1** for E1.

Add a card for T1.

- a** Enter **card add shelf/slot/cardtype linetype type**.

```
zSH> card add 6 linetype ds1
new card-profile 1/6/10214 added, sw-file-name "mxlc24t1e1bond.bin", 1
option: card-line-type ds1
```

- b** Verify the card by entering **slots**:

```
zSH> slots
MXK 819
```

Uplinks

a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)

b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)

Cards

```

1: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
2: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
3: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
4: MXK T1E1-24 Bonded (NOT_PROV)
5: MXK T1E1-24 Bonded (NOT_PROV)
6: MXK T1E1-24 Bonded (RUNNING)
7: MXK ADSL-48-A Bonded (NOT_PROV)

```

c Verify the **card-profile** for the EFM T1E1-24 card, in this case T1:

```

zSH> get card-profile 1/6/10214
card-profile 1/6/10214
sw-file-name: -----> {mxlc24t1e1bond.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {ds1}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}

```

d View card information including the state of the card and how long the card has been running:

```

zSH> slots 6
MXK 819
Type           : MXK T1E1-24 Bonded
Card Version   : 00001
EEPROM Version : 1
Serial #       : 2561128
CLEI Code      : No CLEI
Card-Profile ID : 1/6/10214
Shelf          : 1
Slot           : 6
ROM Version    : MXK 2.1.211
Software Version: MXK 2.1.3.203
State          : RUNNING
Mode           : FUNCTIONAL
Heartbeat check : enabled
Heartbeat last  : TUE JAN 18 18:07:50 2011
Heartbeat resp  : 86230
Heartbeat late  : 0
Hbeat seq error : 0

```

```

Hbeat longest      : 5
Fault reset        : enabled
Power fault mon    : not supported
Uptime             : 8 minutes

```

Add a card for E1.

a Enter **card add slot linetype type**.

```

zSH> card add 5 linetype e1
new card-profile 1/5/10214 added, sw-file-name "mxlc24t1e1bond.bin", 1
option: card-line-type e1

```

b Verify the card by entering **slots**:

```

zSH> slots
MXK 819

```

Uplinks

a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)

b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)

Cards

```

1: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
2: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
3: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
4: MXK T1E1-24 Bonded (NOT_PROV)
5: MXK T1E1-24 Bonded (RUNNING)
6: MXK T1E1-24 Bonded (RUNNING)
7: MXK ADSL-48-A Bonded (NOT_PROV)

```

c Verify the **card-profile** for the EFM T1E1-24 card, in this case T1:

```

zSH> get card-profile 1/5/10214
card-profile 1/5/10214
sw-file-name: -----> {mxlc24t1e1bond.bin}
admin-status: -----> {operational}
upgrade-sw-file-name: ---> {}
upgrade-vers: -----> {}
admin-status-enable: ----> {enable}
sw-upgrade-admin: -----> {reloadcurrrev}
sw-enable: -----> {true}
sw-upgrade-enable: -----> {false}
card-group-id: -----> {0}
hold-active: -----> {false}
weight: -----> {nopreference}
card-line-type: -----> {e1}
card-atm-configuration: -> {notapplicable}
card-line-voltage: -----> {not-used}
maxvpi-maxvci: -----> {notapplicable}
card-init-string: -----> {}
wetting-current: -----> {disabled}
pwe-timing-mode: -----> {none}

```


- d** View card information including the state of the card and how long the card has been running:

```

zSH> slots 5
MXK 819
Type           : MXK T1E1-24 Bonded
Card Version   : 00001
EEPROM Version : 1
Serial #       : 2561133
CLEI Code      : No CLEI
Card-Profile ID : 1/5/10214
Shelf          : 1
Slot           : 5
ROM Version    : MXK 2.1.211
Software Version: MXK 2.1.3.204
State          : RUNNING
Mode           : FUNCTIONAL
Heartbeat check : enabled
Heartbeat last  : TUE JAN 18 22:17:05 2011
Heartbeat resp  : 8533
Heartbeat late  : 0
Hbeat seq error : 0
Hbeat longest   : 69
Fault reset     : enabled
Power fault mon : not supported
Uptime         : 2 hours, 22 minutes

```

Activate a Ds1 interface

The *ds1* interface must be activated before entering other configuration tasks for the EFM T1/E1 card.

Activating a Ds1 interface

- 1** Activate each *ds1* interface by updating the **adminstatus** parameter to *up*.

```

zSH> port up 1-8-1-0/ds1
1-8-1-0/ds1 set to admin state UP

```

```

zSH> port up 1-8-2-0/ds1
1-8-2-0/ds1 set to admin state UP

```

Continue updating each *ds1* interface.

- 2** Proceed to connecting the CPEs after the *ds1* interfaces are active.

The N2N bond groups will be automatically created depending on the number of ports on the CPE that are connected to the EFM T1/E1 card.

View the Ds1 interface

This section describes the Ds1 interface.

To view existing Ds1 profiles on a MXK enter:

```
zSH> list ds1-profile
ds1-profile 1-6-1-0/ds1
ds1-profile 1-6-2-0/ds1
ds1-profile 1-6-3-0/ds1 .
ds1-profile 1-6-4-0/ds1
ds1-profile 1-6-5-0/ds1
ds1-profile 1-6-6-0/ds1
ds1-profile 1-6-7-0/ds1
ds1-profile 1-6-8-0/ds1
ds1-profile 1-6-9-0/ds1
ds1-profile 1-6-10-0/ds1
...
48 entries found.
```

The following example shows the default parameters in the **ds1-profile** for T1 and E1 interfaces.

Default parameters for T1 **ds1-profile**.

```
zSH> get ds1-profile 1-8-1-0/ds1
ds1-profile 1-8-1-0/ds1
line-type: -----> {esf}
line-code: -----> {b8zs}
send-code: -----> {sendnocode}
circuit-id: -----> {ds1}
loopback-config: -----> {nolop}
signal-mode: -----> {none}
fdl: -----> {fdlnone}
dsx-line-length: -----> {dsx0}
line-status_change-trap-enable: ---> {enabled}
channelization: -----> {disabled}
ds1-mode: -----> {csu}
csu-line-length: -----> {csu00}
clock-source-eligible: -----> {eligible}
transmit-clock-source: -----> {throughtiming}
cell-scramble: -----> {true}
coset-polynomial: -----> {true}
protocol-emulation: -----> {network}
signal-type: -----> {loopstart}
ds1-group-number: -----> {0}
line-power: -----> {disabled}
timeslot-assignment: ----->
{0+1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20+21+22+23}
transmit-clock-adaptive-quality: --> {stratum3}
```

Default parameters in the **ds1-profile** for E1.

```
zSH> get ds1-profile 1-5-1-0/ds1
ds1-profile 1-5-1-0/ds1
line-type: -----> {e1crc}
line-code: -----> {hdb3}
send-code: -----> {sendnocode}
circuit-id: -----> {e1}
```

```

loopback-config: -----> {noloop}
signal-mode: -----> {none}
fdl: -----> {fdlnone}
dsx-line-length: -----> {dsx0}
line-status_change-trap-enable: ---> {enabled}
channelization: -----> {disabled}
ds1-mode: -----> {other}
csu-line-length: -----> {csu00}
clock-source-eligible: -----> {eligible}
transmit-clock-source: -----> {throughtiming}
cell-scramble: -----> {true}
coset-polynomial: -----> {true}
protocol-emulation: -----> {network}
signal-type: -----> {loopstart}
ds1-group-number: -----> {0}
line-power: -----> {disabled}
timeslot-assignment: ----->
{0+1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20+21+22+23+24+25+26+27+28+
29+30}
transmit-clock-adaptive-quality: --> {stratum3}

```

Table 210 describes the supported **ds1-profile** parameters.

Table 210: ds1-profile parameters

Parameter	Options
line-type	Indicates the type of Ds1 line implementing this circuit. The type of circuit affects the number of bits per second that the circuit can reasonably carry, as well as the interpretation of the usage and error statistics. Values: esf Extended Super Frame. ami Alternate Mark Inversion. D4 Supported line type for T1. e1Mf : G.704, table 4a, with TS16 multiframing enabled for E1 circuits. e1CrcMf : G.704, table 4b, with TS16 multiframing enabled for E1 circuits. Default: esf for T1 e1 for E1
line-code	Describes the type of Zero Code suppression used on this interface. b8zs: a specific pattern of normal bits and bipolar violations used to replace a sequence of eight zero bits. hdb3: High Density Bipolar of order 3. A code used for E1. Default: b8zs for T1 hdb3 for E1

Table 210: ds1-profile parameters (Continued)

Parameter	Options
send-code	This parameter is used for bit error rate (BER) testing. This variable indicates what type of code is being sent across the Ds1 interface by the device. Setting this variable causes the interface to send the code requested. The values mean: sendnocode: sending looped or normal data sendLineCode: sending a request for a line loopback T1 related sendCodes
circuit-id	This variable contains the transmission vendor's circuit identifier, for the purpose of facilitating troubleshooting. Enter a circuit identifier for the interface, up to 36 characters.
loopback-config	This parameter is used for loopback testing. Variables represent the desired loopback configuration of the Ds1 interface. Agents supporting read/write access should return inconsistentValue in response to a requested loopback state that the interface does not support.
signal mode	Specifies the signaling mode. Default: messageoriented for E1 robbedbit for T1
fdl	Ds1_Profile.fdl Values: other: indicates that a protocol other than one following is used. ansiT1403: refers to the FDL exchange recommended by ANSI. att54016: refers to ESF FDL exchanges. fdlNone: indicates that the device does not use the FDL. Default: fdlNone
dsx-line-length	The length of the DSX WAN interface in feet. This parameter provides information for line build out circuitry. Values: Dsx0 0 feet for the line build out (LBO) setting. Dsx133 133 feet for the LBO. Dsx266 266 feet for the LBO. Dsx399 399 feet for the LBO. Dsx533 533 feet for the LBO. Dsx655 655 feet for the LBO. Default: 0
line-status-change-trap-enable	Specifies whether a trap is generated whenever the line state changes. Values: enabled disabled Default: enabled

Table 210: ds1-profile parameters (Continued)

Parameter	Options
ds1-mode	Type of interface. Values: dsx DS1 interface is DSX csu DS1 interface is CSU other Interface is neither CSU nor DSX Default: csu
csu-line-length	This parameter provides information for line build out circuitry. Values: csu00 0 dB line build out. csu75 -7.5 dB line build out. csu150 -15.0 dB line build out. csu225 -22.5 dB line build out. Default: csu00
transmit-clock-source	Specifies the clock source for the interface. See Chapter 3, MXK Clocking, on page 173 for information about configuring the system clock. (This reference is accurate when incorporating the section into the guide).
clock-source-eligible	Specifies whether clock source is allowed. Default: noteligible for E1 eligible for T1
cell-scramble	Indicates whether ATM cell scrambling is enabled for this interface. Both sides of the connection must agree on whether scrambling is enabled. Values: true Cell scrambling enabled. false Cell scrambling disabled. Default: true
coset-polynomial	Indicates whether the coset polynomial is used to calculate the ATM header error control (HEC) value. Both sides of the connection must agree on the method of calculating the HEC value. Values: true The coset polynomial is used to calculate the HEC value. false The coset polynomial is not used to calculate the HEC value. Default: true
protocol-emulation	Indicates whether the device is acting as network-side or CPE with respect to this Ds1. Values: network cpe Default: network

Table 210: ds1-profile parameters (Continued)

Parameter	Options
signal-type	The signaling type of the FXS interfaces within this Ds1. Values: loopstart groundstart Default: loopstart
ds1-group-number	The group index this Ds1 belongs to.
line-power	Enable or disable line power for the Ds1 interface. Values: enabled disabled Default: disabled
timeslot-assignment	This table entry is a bit field indicating which timeslots in a Ds1 are used (or assigned). Default for Ds1 based card: 1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+0+0+0+0+0+0+0 Default for E1 based card: 1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1
transmit-clock-adaptive-quality	Determines sync drift when operating in PWE adaptive mode. Values reflect ANSI Standard T1.101 reference clock quality. Values: stratum1 stratum3 stratum3e stratum4 Default: stratum3

Updating a Ds1 interface

The default values are appropriate for most applications. If you need to change them, update the **ds1-profile** for the interface:

```

ZSH> update ds1-profile 1-8-1-0/ds1
ds1-profile 1-8-1-0/ds1
Please provide the following: [q]uit.
line-type: -----> {esf}:
line-code: -----> {b8zs}:
send-code: -----> {sendnocode}:
circuit-id: -----> {ds1}:
loopback-config: -----> {nolloop}:
signal-mode: -----> {none}:

```

```

fdl: -----> {fdlnone}:
dsx-line-length: -----> {dsx0}:
line-status_change-trap-enable: ---> {enabled}:
channelization: -----> {disabled}:
dsl-mode: -----> {csu}:
csu-line-length: -----> {csu00}:
clock-source-eligible: -----> {eligible}:
transmit-clock-source: -----> {throughtiming}: localtiming
cell-scramble: -----> {true}:
coset-polynomial: -----> {true}:
protocol-emulation: -----> {cpe}:
signal-type: -----> {loopstart}:
dsl-group-number: -----> {0}:
line-power: -----> {disabled}:
timeslot-assignment: ----->
{0+1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20+21+22+23}:
transmit-clock-adaptive-quality: --> {stratum3}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

Net-to-net bonding

This section describes N2N (net-to-net) bonding:

- [EFM auto bonding, page 1484](#)
- [Display bond groups, page 1484](#)
- [Create bond groups from the CLI, page 1485](#)
- [Delete bond groups, page 1487](#)

EFM (Ethernet in the First Mile) extends Ethernet signaling between the MXK-EFM-T1/E1-24 card and the CPEs.

By default, all ports are configured in N2N bond groups.

EFM auto bonding

EFM discovery automatically groups ports that are connected to the same CPE to create a dynamic bond group utilizing automatic creation bond group numbers (25–505). The valid ranges for all EFM bond groups are:

- 25–505 for CLI created bond groups.
- 25–505 for ZMS create bond groups.
- Automatic creation starts from 505 and goes down sequentially as the bond groups are created.

EFM T1/E1 cards on the MXK support up to 24 bond groups. Each bond group can have a maximum of eight members.

The number of bond groups on a EFM T1/E1 card depends on the number of ports that exist on the CPE devices connected to the EFM T1/E1 card. For example, a EFM T1/E1 card connected to six four-port CPE devices would have six bond groups.

Display bond groups

Bond groups can be displayed for all existing groups, a specific group, a specific slot, or link.

To display all configured bond groups:

```
zSH> bond show all
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
5	479	n2nbond	ACT	bond-0479	-
5	480	n2nbond	ACT	bond-0480	-
5	481	n2nbond	ACT	bond-0481	-
5	482	n2nbond	ACT	bond-0482	-
5	483	n2nbond	ACT	bond-0483	-
5	484	n2nbond	ACT	bond-0484	-
6	499	n2nbond	ACT	bond-0499	-
6	488	n2nbond	ACT	bond-0488	-

6	489	n2nbond	ACT	bond-0489	-
6	490	n2nbond	ACT	bond-0490	-
6	491	n2nbond	ACT	bond-0491	-
6	492	n2nbond	ACT	bond-0492	-
6	493	n2nbond	ACT	bond-0493	-
6	494	n2nbond	ACT	bond-0494	-
6	495	n2nbond	ACT	bond-0495	-
6	496	n2nbond	ACT	bond-0496	-
6	498	n2nbond	ACT	bond-0498	-

To display a specific bond group:

```
zSH> bond show group bond-0495/n2nbond
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
6	495	n2nbond	ACT	bond-0495	-
Group Members					
Slot	Port	Type	State	Name	Desc
6	5	ds1	ACT	1-6-5-0	-
6	8	ds1	ACT	1-6-8-0	-
6	6	ds1	ACT	1-6-6-0	-
6	7	ds1	ACT	1-6-7-0	-

To display bond groups by slot:

```
zSH> bond show slot 6
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
6	499	n2nbond	ACT	bond-0499	-
6	488	n2nbond	ACT	bond-0488	-
6	489	n2nbond	ACT	bond-0489	-
6	490	n2nbond	ACT	bond-0490	-
6	491	n2nbond	ACT	bond-0491	-
6	492	n2nbond	ACT	bond-0492	-
6	493	n2nbond	ACT	bond-0493	-
6	494	n2nbond	ACT	bond-0494	-
6	495	n2nbond	ACT	bond-0495	-
6	496	n2nbond	ACT	bond-0496	-
6	498	n2nbond	ACT	bond-0498	-

To display bond groups for a specific link:

```
zSH> bond show link 1-6-1-0/ds1
```

Bond Groups					
Slot	GrpId	Type	State	Name	Desc
6	499	n2nbond	ACT	bond-0499	-
Group Members					
Slot	Port	Type	State	Name	Desc
6	1	ds1	ACT	1-6-1-0	-

Create bond groups from the CLI

When you need to create bond groups from the CLI, first create the N2N bond group, then add the links to that group before connecting the CPE.

The MXK T1/E1 connector has 24 EFM T1/E1 ports and supports up to 24 bond groups.

Creating bond groups from the CLI

- 1 Enter slots to verify the location of the EFM T1/E1 card.

```
zSH> slots
MXK 819
```

Uplinks

a:*MXK TWO TENGIGE EIGHT GIGE (RUNNING+TRAFFIC)

b: MXK TWO TENGIGE EIGHT GIGE (RUNNING)

```
Cards
1: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
2: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
3: MXK GSHDSL-24 Bonded/with NTP (RUNNING)
4: MXK T1E1-24 Bonded (RUNNING)
5: MXK T1E1-24 Bonded (RUNNING)
6: MXK T1E1-24 Bonded (RUNNING)
```

- 2 Enter **bond add group** *bond-bondGroupName/type* to create a bond group with *port* designating the group ID of the bond group.



Note: In the case of bond group commands, *port* refers to the group ID of the bond group.

```
zSH> bond add group 1-4-101-0/n2nbond
Bond group - bond-0101/n2nbond - was successfully created.
```

When entering a bond group *interface/n2nbond*, check to see which interface is actually created. If the bond group already exists, the system creates the interface with a system assigned value.

- 3 Add a single bond group member to the bond group by entering **bond add member** *bond-bondGroupName/type* with the interface and type of the bond group followed by the interface and type of the group member to be added.

```
zSH> bond add member bond-0101/n2nbond 1-4-1-0/ds1
```

Add several bond group members to the bond group.

```
zSH> bond add member bond-0101/n2nbond 1-4-2-0/ds1 1-4-3-0/ds1 1-4-4-0/ds1
```

- 4 Enter **bond show group** *bond-bondGroupName/type* to verify the members of the bond group.

```
zSH> bond show group bond-0101/n2nbond
Bond Groups
Slot  GrpId  Type      State  Name      Desc
```

4	101	n2nbond	OOS	bond-0101	-
Group Members					
Slot	Port	Type	State	Name	Desc
4	1	ds1	OOS	1-4-1-0	-
4	4	ds1	OOS	1-4-4-0	-
4	3	ds1	OOS	1-4-3-0	-
4	2	ds1	OOS	1-4-2-0	-

Delete bond groups

Bond groups can be deleted by individual member or entire group.

```
zSH> bond delete member bond-0101/n2nbond 1-4-1-0/ds1
```

Or:

```
zSH> bond delete group bond-0101/n2nbond
```

Bond group statistics and port statistics

The MXK provides the ability to view statistics for both each port associated with a link in a bond group and an individual bond group.

- [Display statistics for an T1/E1 port, page 1488](#)
- [Display statistics for a bond group, page 1491](#)

Display statistics for an T1/E1 port

EFM bonding fragments packets across multiple lines so that packet counts for each EFM port indicates the number of EFM packet fragments for that port. At the physical port level, unicast packet counts show the number of packet fragments for that port. Octets for the physical port include all bytes received, including those from errored packet fragments and protocol overhead.

```
zSH> ds1stat 1-4-1-0/ds1
Line Information:
-----
Alarm Status.....1
                                ->No Alarm
Line Type.....Ext Super Frame
Ds1 Mode.....CSU
Signal Type.....Loop start
Time Elapsed.....594
LineStatusLastChange.....32421900
Transmit Clock Source.....Loop Timing
Loopback Status.....1
                                ->No Loopback
*****      Pmon Statistics of Line 371      *****
INT   PCV   LCV   LES   CSS   ES    BES   SES   SEFS  DM    UAS
-----
Near-End Current Interval Stats:
-----
--    0     0     0     0     0     0     0     0     0     0     0
Near-End Interval Stats:
-----
Retrieving data in progress ...
Done.
1     0     0     0     0     0     0     0     0     0     0
2     0     0     0     0     0     0     0     0     0     228
3     0     0     0     0     0     0     0     0     0     900
4     0     0     0     0     0     0     0     0     0     425
5     0     0     0     0     0     0     0     0     0     338
6     0     0     0     0     0     0     0     0     0     236
7     0     0     0     0     0     0     0     0     0     0
8     0     0     0     0     0     0     0     0     0     0
9     0     0     0     0     0     0     0     0     0     0
10    0     0     0     0     0     0     0     0     0     0
11    0     0     0     0     0     0     0     0     0     0
12    0     0     0     0     0     0     0     0     0     0
```

```

13      0      0      0      0      0      0      0      0      0      0
....
94      0      0      0      0      0      0      0      0      0      0
95      0      0      0      0      0      0      0      0      0      0
96      0      0      0      0      0      0      0      0      0      0
Near-End Total Stats:
-----
--      0      0      0      0      0      0      0      0      0      2127
*****                                     *****
                                     End

```

[Table 211](#) describes the parameters for the **ds1stat** command.

Table 211: ds1stat command display fields

Field	Description
Alarm status	Status of the alarm.
Line type	This variable indicates the variety of Ds1 line implementing this circuit. The type of circuit affects the number of bits per second that the circuit can reasonably carry, as well as the interpretation of the usage and errors statistics. Supported values: esf Extended Super Frame Ds1 (T1.107) d4 AT&T D4 format Ds1 (T1.107) e1 ITU-T Recommendation G.704 e1-CRC ITU-T Recommendation G.704
Ds1 mode	other wan is not in csu or dsx mode. dsx T1 wan is in dsx mode. csu T1 wan is in csu mode. Default setting is csu for T1 platforms and other for E1.
Signal type	The signaling type of the FXS interfaces within this Ds1. Values: loopStart groundStart Default: loopStart
Time Elapsed	The number of seconds that have elapsed since the beginning of the near end current error-measurement period. If, for some reason, such as an adjustment in the system's time-of-day clock, the current interval exceeds the maximum value, the agent will return the maximum value.

Table 211: ds1stat command display fields (Continued)

Field	Description
LineStatusLast Change	The value of MIB II's sysUpTime object at the time this Ds1 entered its current line status state. If the current state was entered prior to the last re-initialization of the proxy-agent, then this object contains a zero value.
Transmit clock source	Reflects the source of Transmit Clock. loopTiming indicates that the recovered receive clock is used as the transmit clock. Only set if this Ds1 has zhoneClockSourceEligibility equal to 'eligible' and the state of that Ds1 is 'up' and the Ds1 resource provider decided this Ds1 is to provide the transmit clock for all the Ds1s. Only one Ds1 will have this set per platform. Values: loopTiming localTiming throughTiming Default: throughTiming
LoopbackStatus	This variable represents the current state of the loopback on the Ds1 interface. It contains information about loopbacks established by a local manager and remotely from the far end. This status is combination of loopbackConfig, and sendCode options as this status represents the local as well as far loopbacks. The various positions are: noLoopback nearEndLineLoopback nearEndOtherLoopback nearEndLocalLoopback farEndPayloadLoopback farEndLineLoopback
INT (Interval)	Intervals are 900 second (15 minute) buckets. You can gather up to 96 intervals (24 hours) of history.
PCV (Path Coding Violations)	Frame synchronization errors in D4 and E1- no CRC formats; May also be a CRC error in ESF and E1 - CRC formats.
LCV (Line Code Violations)	An LCV is the occurrence of a Bipolar Violation (BPV) or Excessive Zeroes (EXZ) error event). A BPV error event occurs when two pulses of the same polarity occur without the opposite polarity occurring. With T1 pulses (represents ONE, no pulse represents ZERO) alternate polarity. If two pulses of the same polarity are received in succession, either bits were added or deleted from the signal. EXZ = If too many zeros (no pulse) are received in succession, this event can cause receiving equipment to lose synchronization with the sending equipment.

Table 211: ds1stat command display fields (Continued)

Field	Description
LES Line Errored Seconds	The number of Line Errored Seconds (when one or more LCV violation events are detected in a second).
CSS (Controlled Slip Seconds)	Controlled slip seconds when at least one controlled slip occurs. A controlled slip is when the detected error is in deletion or replication of a frame.
ES (Errored Seconds)	An Errored Second has one or more Path Code Violation, one or more Out of Frame defects, one or more Controlled Slip events, or a detected Alarm Indication Signal (AIS) defect. AIS defects are sent to the receiver when a transmission interruption is detected from the device transmitting the signal or a device upstream which sends the signal which may be forwarded.
BES (Bursty Errored Seconds)	The number of Bursty Error Seconds with 2 to 319 PCV error events, but no severely error frame defects and no detected incoming AIS defects.
SES (Severely Errored Seconds)	A Severely Errored Second is a second with 320 or more Path Code Violation Error Events OR one or more Out of Frame (OOF) defects OR a detected AIS defects. Transmission performance is significantly degraded. For T1 links, an Out of Frame defect is declared when the receiver detects two or more framing errors within a 3 msec period for ESF signals and 0.75 msec for D4 signals, or two or more errors out of five or fewer consecutive framing-bits. For E1 links, an Out Of Frame defect is declared when three consecutive frame alignment signals have been received with an error.
SEFS (Severely Errored Framing Seconds)	SEFS are seconds with one or more Out of Frame defects or a detected AIS defect.
DM (Degraded Minutes)	Degraded minutes are a range of errors per minute. Degraded Minutes are when the estimated error rate exceeds 1E-6 per minute, but does not exceed 1E-3 errors per minute.
UAS (Unavailable Seconds)	The Ds1 interface is considered unavailable when 10 contiguous SESs occur OR the onset of a failure condition (see RFC 1406 for a list of failure states).

Display statistics for a bond group

Bond group statistics can be displayed for a bond group interface.

```

zSH> bond stats bond-0495/n2nbond
***** Bond group statistics *****
                Group Info
Slot      GrpId      Interface Name
   6        495      bond-0495/n2nbond
AdminStatus  OperStatus  Bandwidth    Last Change
   UP          UP        6144000      1.01:19:09
Threshold Alarm Config

```

disabled

Group Members				
Port	Admin	Oper	Bandwidth	Interface Name
6/5	UP	UP	1536000	1-6-5-0/ds1
6/8	UP	UP	1536000	1-6-8-0/ds1
6/6	UP	UP	1536000	1-6-6-0/ds1
6/7	UP	UP	1536000	1-6-7-0/ds1

Statistics (Received)

Octets	61140462
Ucast	882357
Mcast	53061
Bcast	6
Discards	0
Errors	0

Statistics (Transmitted)

Octets	1306645829
Ucast	881612
Mcast	60772
Bcast	291
Discards	0

EFM T1/E1 24-port cables

This sections describes the EFM T1/E1 cables:

- [MALC-CBL-T1/E1-2-45DEG](#), page 1493
- [Blunt cables](#), page 1497

Cables and pinouts are the same for the MALC EFM T1/E1 and the MXK EFM T1/E1 card.

MALC-CBL-T1/E1-2-45DEG

[Figure 177](#) shows the MXK EFM T1/E1 24-port bonding cable (MALC-CBL-T1/E1-24-45DEG). [Table 215 on page 1496](#) lists the pinouts.

Figure 177: MXK T1/E1 24 port cable

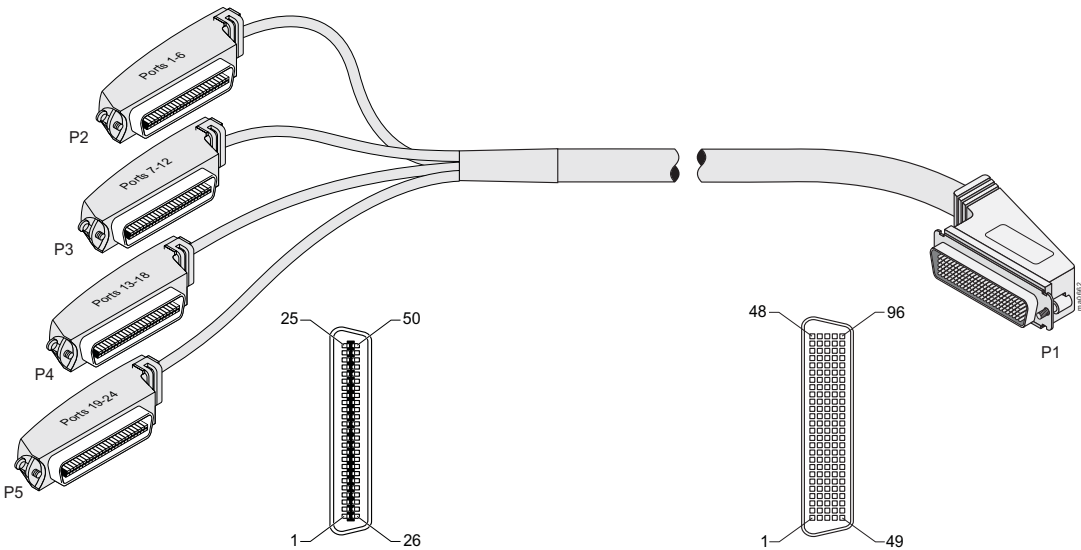


Table 212: Port-Pair Detail Ports 1-6 (P1 to P2)

Port	Pair	Signal	Color	From	To
1	1	TX 1 Ring	BLU/WHT	P1-1	P2-1
		TX 1 Tip	WHT/BLU	P1-2	P2-26
	2	RX 1 Ring	ORG/WHT	P1-3	P2-27
		RX 1 Tip	WHT/ORG	P1-4	P2-2
2	3	TX 2 Ring	GRN/WHT	P1-5	P2-5
		TX 2 Tip	WHT/GRN	P1-6	P2-30
	4	RX 2 Ring	BRN/WHT	P1-7	P2-31

Table 212: Port-Pair Detail Ports 1-6 (P1 to P2) (Continued)

Port	Pair	Signal	Color	From	To
3	5	RX 2 Tip	WHT/BRN	P1-8	P2-6
		TX 3 Ring	SLT/WHT	P1-9	P2-9
		TX 3 Tip	WHT/SLT	P1-10	P2-34
4	6	RX 3 Ring	BLU/RED	P1-11	P2-35
		RX 3 Tip	RED/BLU	P1-12	P2-10
		TX 4 Ring	ORG/RED	P1-13	P2-13
5	7	TX 4 Tip	RED/ORG	P1-14	P2-38
		RX 4 Ring	GRN/RED	P1-15	P2-39
		RX 4 Tip	RED/GRN	P1-16	P2-14
6	9	TX 5 Ring	BRN/RED	P1-17	P2-17
		TX 5 Tip	RED/BRN	P1-18	P2-42
		RX 5 Ring	SLT/RED	P1-19	P2-43
6	10	RX 5 TIP	RED/SLT	P1-20	P2-18
		TX 6 Ring	BLU/BLK	P1-21	P2-21
		TX 6 Tip	BLK/BLU	P1-22	P2-46
6	11	RX 6 Ring	ORG/BLK	P1-23	P2-47
		RX 6 TIP	BLK/ORG	P1-24	P2-22

Table 213: Port-Pair Detail Ports (P1 to P3) 7-12

Port	Pair	Signal	Color	From	To
7	13	TX 7 Ring	BLU/WHT	P1-25	P3-1
		TX 7 Tip	WHT/BLU	P1-26	P3-26
	14	RX 7 Ring	ORG/WHT	P1-27	P3-27
		RX 7 Tip	WHT/ORG	P1-28	P3-2
8	15	TX 8 Ring	GRN/WHT	P1-29	P3-5
		TX 8 Tip	WHT/GRN	P1-30	P3-30
	16	RX 8 Ring	BRN/WHT	P1-31	P3-31
		RX 8 Tip	WHT/BRN	P1-32	P3-6

Table 213: Port-Pair Detail Ports (P1 to P3) 7-12 (Continued)

Port	Pair	Signal	Color	From	To
9	17	TX 9 Ring	SLT/WHT	P1-33	P3-9
		TX 9 Tip	WHT/SLT	P1-34	P3-34
	18	RX 9 Ring	BLU/RED	P1-35	P3-35
		RX 9 Tip	RED/BLU	P1-36	P3-10
10	19	TX 10 Ring	ORG/RED	P1-37	P3-13
		TX 10 Tip	RED/ORG	P1-38	P3-38
	20	RX 10 Ring	GRN/RED	P1-39	P3-39
		RX 10 Tip	RED/GRN	P1-40	P3-14
11	21	TX 11 Ring	BRN/RED	P1-41	P3-17
		TX 11 Tip	RED/BRN	P1-42	P3-42
	22	RX 11 Ring	SLT/RED	P1-43	P3-43
		RX 11 Tip	RED/SLT	P1-44	P3-18
12	23	TX 12 Ring	BLU/BLK	P1-45	P3-21
		TX 12 Tip	BLK/BLU	P1-46	P3-46
	24	RX 12 Ring	ORG/BLK	P1-47	P3-47
		RX 12 Tip	BLK/ORG	P1-48	P3-22

Table 214: Port-Pair Detail Ports (P1 to P4)13-18

Port	Pair	Signal	Color	From	To
13	25	TX 13 Ring	BLU/WHT	P1-49	P4-1
		TX 13 Tip	WHT/BLU	P1-50	P4-26
	26	RX 13 Ring	ORG/WHT	P1-51	P4-27
		RX 13 Tip	WHT/ORG	P1-52	P4-2
14	27	TX 14 Ring	GRN/WHT	P1-53	P4-5
		TX 14 Tip	WHT/GRN	P1-54	P4-30
	28	RX 14 Ring	BRN/WHT	P1-55	P4-31
		RX 14 Tip	WHT/BRN	P1-56	P4-6

Table 214: Port-Pair Detail Ports (P1 to P4)13-18 (Continued)

Port	Pair	Signal	Color	From	To
15	29	TX 15 Ring	SLT/WHT	P1-57	P4-9
		TX 15 Tip	WHT/SLT	P1-58	P4-34
	30	RX 15 Ring	BLU/RED	P1-59	P4-35
		RX 15 Tip	RED/BLU	P1-60	P4-10
16	31	TX 16 Ring	ORG/RED	P1-61	P4-13
		TX 16 Tip	RED/ORG	P1-62	P4-38
	32	RX 16 Ring	GRN/RED	P1-63	P4-39
		RX 16 Tip	RED/GRN	P1-64	P4-14
17	33	TX 17 Ring	BRN/RED	P1-65	P4-17
		TX 17 Tip	RED/BRN	P1-66	P4-42
	34	RX 17 Ring	SLT/RED	P1-67	P4-43
		RX 17 Tip	RED/SLT	P1-68	P4-18
18	35	TX 18 Ring	BLU/BLK	P1-69	P4-21
		TX 18 Tip	BLK/BLU	P1-70	P4-46
	36	RX 18 Ring	ORG/BLK	P1-71	P4-47
		RX 18 Tip	BLK/ORG	P1-72	P4-22

Table 215: Port-Pair Detail Ports (P1 to P5) 19-24

Port	Pair	Signal	Color	From	To
19	37	TX 19 Ring	BLU/WHT	P1-73	P5-1
		TX 19 Tip	WHT/BLU	P1-74	P5-26
	38	RX 19 Ring	ORG/WHT	P1-75	P5-27
		RX 19 Tip	WHT/ORG	P1-76	P5-2
20	39	TX 20 Ring	GRN/WHT	P1-77	P5-5
		TX 20 Tip	WHT/GRN	P1-78	P5-30
	40	RX 20 Ring	BRN/WHT	P1-79	P5-31
		RX 20 Tip	WHT/BRN	P1-80	P5-6

Table 215: Port-Pair Detail Ports (P1 to P5) 19-24 (Continued)

Port	Pair	Signal	Color	From	To
21	41	TX 21 Ring	SLT/WHT	P1-81	P5-9
		TX 21 Tip	WHT/SLT	P1-82	P5-34
	42	RX 21 Ring	BLU/RED	P1-83	P5-35
		RX 21 Tip	RED/BLU	P1-84	P5-10
22	43	TX 22 Ring	ORG/RED	P1-85	P5-13
		TX 22 Tip	RED/ORG	P1-86	P5-38
	44	RX 22 Ring	GRN/RED	P1-87	P5-39
		RX 22 Tip	RED/GRN	P1-88	P5-14
23	45	TX 23 Ring	BRN/RED	P1-89	P5-17
		TX 23 Tip	RED/BRN	P1-90	P5-42
	46	RX 23 Ring	SLT/RED	P1-91	P5-43
		RX 23 Tip	RED/SLT	P1-92	P5-18
24	47	TX 24 Ring	BLU/BLK	P1-93	P5-21
		TX 24 Tip	BLK/BLU	P1-94	P5-46
	48	RX 24 Ring	ORG/BLK	P1-95	P5-47
		RX 24 Tip	BLK/ORG	P1-96	P5-22

Blunt cables

Several blunt-end MALC-EFM-T1/E1-24 card cable options are supported.

- MALC-CBL-ADSL-48-100FT-BLUNT
- MALC-CBL-ADSL-48-350FT-BLUNT
- MALC-CBL-ADSL-48-30FT-BLUNT
- MALC-CBL-ADSL-48-15FT-BLUNT

The following tables list the blunt cable pinouts.

Table 216: Pinout for high density connector to blunt end cable

Port	Pair	Signal	Color	Form	Binder Group
1	1	TX 1 Ring	Blue/White	P1-1	1 (Blue)
		TX 1 Tip	White/Blue	P1-2	
	2	RX 1 Ring	Orange/White	P1-3	
		TX 1 Tip	White/Orange	P1-4	
2	3	TX 2 Ring	Green/White	P1-5	
		TX 2 Tip	White/Green	P1-6	
	4	RX 2 Ring	Brown/White	P1-7	
		TX 2 Tip	White/Brow	n P1-8	
3	5	TX 3 Ring	Slate/White	P1-9	
		TX 3 Tip	White/Slate	P1-10	
	6	RX 3 Ring	Blue/Red	P1-11	
		TX 3 Tip	Red/Blue	P1-12	
4	7	TX 4 Ring	Orange/Red	P1-13	
		TX 4 Tip	Red/Orange	P1-14	
	8	RX 4 Ring	Green/Red	P1-15	
		TX 4 Tip	Red/Green	P1-16	
5	9	TX 5 Ring	Brown/Red	P1-17	
		TX 5 Tip	Red/Brown	P1-18	
	10	RX 5 Ring	Slate/Red	P1-19	
		TX 5 Tip	Red/Slate	P1-20	
6	11	TX 6 Ring	Blue/Black	P1-21	
		TX 6 Tip	Black/Blue	P1-22	
	12	RX 6 Ring	Orange/Black	P1-23	
		TX 6 Tip	Black/Orange	P1-24	

Table 217: Pinout for high density connector to blunt end cable

Port	Pair	Signal	Color	Form	Binder Group
7	13	TX 7 Ring	Blue/White	P1-25	2 (Orange)
		TX 7 Tip	White/Blue	P1-26	
	14	RX 7 Ring	Orange/White	P1-27	
		TX 7 Tip	White/Orange	P1-28	
8	15	TX 8 Ring	Green/White	P1-29	
		TX 8 Tip	White/Green	P1-30	
	16	RX 8 Ring	Brown/White	P1-31	
		TX 8 Tip	White/Brown	P1-32	
9	17	TX 9 Ring	Slate/White	P1-33	
		TX 9 Tip	White/Slate	P1-34	
	18	RX 9 Ring	Blue/Red	P1-35	
		TX 9 Tip	Red/Blue	P1-36	
10	19	TX 10 Ring	Orange/Red	P1-37	
		TX 10 Tip	Red/Orange	P1-38	
	20	RX 10 Ring	Green/Red	P1-39	
		TX 10 Tip	Red/Green	P1-40	
11	21	TX 11 Ring	Brown/Red	P1-41	
		TX 11 Tip	Red/Brown	P1-42	
	22	RX 11 Ring	Slate/Red	P1-43	
		TX 11 Tip	Red/Slate	P1-44	
12	23	TX 12 Ring	Blue/Black	P1-45	
		TX 12 Tip	Black/Blue	P1-46	
	24	RX 12 Ring	Orange/Black	P1-47	
		TX 12 Tip	Black/Orange	P1-48	

Table 218: Pinout for high density connector to blunt end cable

Port	Pair	Signal	Color	Form	Binder Group
13	25	TX 13 Ring	Blue/White	P1-49	3 (Green)
		TX 13 Tip	White/Blue	P1-50	
	26	RX 13 Ring	Orange/White	P1-53	
		TX 13 Tip	White/Orange	P1-52	
14	27	TX 14 Ring	Green/White	P1-55	
		TX 14 Tip	White/Green	P1-54	
	28	RX 14 Ring	Brown/White	P1-57	
		TX 14 Tip	White/Brown	P1-56	
15	29	TX 15 Ring	Slate/White	P1-59	
		TX 15 Tip	White/Slate	P1-58	
	30	RX 15 Ring	Blue/Red	P1-61	
		TX 15 Tip	Red/Blue	P1-60	
16	31	TX 16 Ring	Orange/Red	P1-63	
		TX 16 Tip	Red/Orange	P1-62	
	32	RX 16 Ring	Green/Red	P1-65	
		TX 16 Tip	Red/Green	P1-64	
17	33	TX 17 Ring	Brown/Red	P1-67	
		TX 17 Tip	Red/Brown	P1-66	
	34	RX 17 Ring	Slate/Red	P1-69	
		TX 17 Tip	Red/Slate	P1-68	
18	35	TX 18 Ring	Blue/Black	P1-71	
		TX 18 Tip	Black/Blue	P1-70	
	36	RX 18 Ring	Orange/Black	P1-73	
		TX 18 Tip	Black/Orange	P1-72	

Table 219: Pinout for high density connector to blunt end cable

Port	Pair	Signal	Color	Form	Binder Group
19	37	TX 19 Ring	Blue/White	P1-73	4 (Brown)
		TX 19 Tip	White/Blue	P1-74	
	38	RX 19 Ring	Orange/White	P1-75	
		TX 19 Tip	White/Orange	P1-76	
20	39	TX 20 Ring	Green/White	P1-77	
		TX 20 Tip	White/Green	P1-78	
	40	RX 20 Ring	Brown/White	P1-79	
		TX 20 Tip	White/Brown	P1-80	
21	41	TX 21 Ring	Slate/White	P1-81	
		TX 21 Tip	White/Slate	P1-82	
	42	RX 21 Ring	Blue/Red	P1-83	
		TX 21 Tip	Red/Blue	P1-84	
22	43	TX 22 Ring	Orange/Red	P1-85	
		TX 22 Tip	Red/Orange	P1-86	
	44	RX 22 Ring	Green/Red	P1-87	
		TX 22 Tip	Red/Green	P1-88	
23	45	TX 23 Ring	Brown/Red	P1-89	
		TX 23 Tip	Red/Brown	P1-90	
	46	RX 23 Ring	Slate/Red	P1-91	
		TX 23 Tip	Red/Slate	P1-92	
24	47	TX 24 Ring	Blue/Black	P1-93	
		TX 24 Tip	Black/Blue	P1-94	
	48	RX 24 Ring	Orange/Black	P1-95	
		TX 24 Tip	Black/Orange	P1-96	

Tests on the EFM T1/E1 card

This section describes testing on the EFM T1/E1 card:

- [T1/E1 Test Access, page 1502](#)
- [Bit Error Rate Testing \(BERT\), page 1502](#)

T1/E1 Test Access

The T1/E1 test access feature provides the ability to route the 4-wires of a T1/E1 circuit from a MXK T1/E1 line card to the test access ports on a TAC card to achieve look-out testing of a T1/E1 circuit using an external T1 test set.

Perform a look-out test on a T1/E1 circuit MXK T1/E1 line card by updating the **mtac-profile**. Specify **ds1** as the line type in the **ifIndex** parameter, as shown below:

```
zSH> update mtac-profile 1
Please provide the following: [q]uit.
ifIndex: ---> {0/0/0/0/0} 1/3/1/0/ds1
test_mode: -> {mtacmodenone} mtacmodelookout
Bad enum value 0: field test_id
test_id: ---> {NONE(0)}:
param1: ----> {0}:
param2: ----> {0}:
param3: ----> {0}:
param4: ----> {0}:
param5: ----> {0}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```



Note: The **test_mode** parameter in the mtac-profile profile will automatically revert to its default value (**mtacmodenone**) after 5-minutes, which prevents a user from forgetting to reset it. Because you have 5-minutes to perform the test, you should have your external T1/E1 test equipment (e.g. external BERT testers) configured and connected to the MTAC/TAC before updating the **mtac-profile**.

Bit Error Rate Testing (BERT)

The **send-code** parameter in the **ds1-profile** controls loopbacks and BER tests on the T1 interface. The following table describes the BERT options.

Note that the MXK-EFM-T1/E1-24 line card has different testing options, but *only when operating in T1 mode*. See [BERT for T1 EFM, page 1504](#) for more information.

Parameter	Description
send-code	Indicates what type of code is being sent across the Ds1 interface by the device. Setting this parameter causes the interface to send the requested code. Values: sendQRSSPattern Sends a Quasi-Random Signal Source (QRSS) test pattern. send511Pattern Sends a 511 bit fixed test pattern. send3in24Pattern Sends a fixed test pattern of 3 bits set in 24. send2047Pattern Sends 2047 test pattern. send1in2Pattern Sends alternate one, zero pattern

Activating a BER test



Note: BER tests disrupt traffic on the interface.

1 Update the **ds1-profile** to specify the BERT pattern:

```

zSH> update ds1-profile 1-4-1-0/ds1
ds1-profile 1-4-1-0/ds1
Please provide the following: [q]uit.
line-type: -----> {esf}:
line-code: -----> {b8zs}:
send-code: -----> {sendnocode}: sendqrsspattern
circuit-id: -----> {ds1}:
loopback-config: -----> {nolooop}:
signal-mode: -----> {none}:
fdl: -----> {fdlnone}:
dsx-line-length: -----> {dsx0}:
line-status_change-trap-enable: ---> {enabled}:
channelization: -----> {disabled}:
ds1-mode: -----> {csu}:
csu-line-length: -----> {csu00}:
clock-source-eligible: -----> {eligible}:
transmit-clock-source: -----> {throughtiming}:
cell-scramble: -----> {true}:
coset-polynomial: -----> {true}:
protocol-emulation: -----> {network}:
signal-type: -----> {loopstart}:
ds1-group-number: -----> {0}:
line-power: -----> {disabled}:
timeslot-assignment: ----->
{0+1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20+21+22+23}:
transmit-clock-adaptive-quality: --> {stratum3}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

2 To end a BER test:

```

zSH> update ds1-profile 1-4-1-0/ds1
ds1-profile 1-4-1-0/ds1
Please provide the following: [q]uit.
line-type: -----> {esf}:
line-code: -----> {b8zs}:
send-code: -----> {sendqrsspattern}: sendnocode
circuit-id: -----> {ds1}:
loopback-config: -----> {nolooop}:
signal-mode: -----> {none}:
fdl: -----> {fdlnone}:
dsx-line-length: -----> {dsx0}:
line-status_change-trap-enable: ---> {enabled}:
channelization: -----> {disabled}:
ds1-mode: -----> {csu}:
csu-line-length: -----> {csu00}:
clock-source-eligible: -----> {eligible}:
transmit-clock-source: -----> {throughtiming}:
cell-scramble: -----> {true}:
coset-polynomial: -----> {true}:
protocol-emulation: -----> {network}:
signal-type: -----> {loopstart}:
ds1-group-number: -----> {0}:
line-power: -----> {disabled}:
timeslot-assignment: ----->
{0+1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20+21+22+23}:
transmit-clock-adaptive-quality: --> {stratum3}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

BERT for T1 EFM

The MXK supports the ability to perform Integrated Bit Error Rate Testing (BERT) for T1 circuits on a T1/E1 EFM line card. This test supports *qrss*, *prbs20* and *prbs15* test patterns to simplify the process of troubleshooting T1 facility problems without the use of external BER test equipment.

The BER testing is only supported in T1 mode.

BER testing can be used with pairs of devices, with one device at each end of a transmission link, or singularly with a loopback at the remote end, so the originating device receives the transmitted BER test.

Before running the actual test, a loop code is sent to the far end device, so the device will know how to respond to the BER test. There are three supported codes: *lineloop*, *payloadloop* and *noloop*.

There are three types of test patterns: *qrss*, *prbs20* and *prbs15*.

The BER test can be run for a selectable amount of time (between 10-300 seconds).

To run a BER Test for T1 EFM use the **ds1bert start** command:

```
ds1bert start <interface> <type> <duration> <loop-up>
```

<interface> can take three forms:

- *ifIndex* (e.g. 432)
- *name/type* (e.g. 1-4-1-0/ds1)
- *shelf/slot/port/subport/type* (e.g. 1/4/1/0/ds1)

<type> is the type of pattern to use for the BERT test on the ds1:

- *qrss* (Quasi Random Signal Source)

QRSS is the most common testing pattern for T1 maintenance and installation. QRSS simulates customer traffic on the circuit and stresses timing recovery, ALBO (automatic line build out) and equalizer circuits.

QRSS is a pseudo-random binary sequencer which generates every combination of a 20-bit word, and repeats every 1,048,575 bits. Consecutive zeros are suppressed to no more than fourteen (14) with a maximum of 20 consecutive ones in the pattern. It contains both high and low density sequences and is a modified version of the PRBS20 test pattern.

The QRSS pattern can be used framed or unframed and will force a B8ZS code in circuits optioned as B8ZS.

- *prbs20* (Pseudo Random Bit Sequence-20)

PRBS20 is a PRBS pattern where a maximum of 19 consecutive zeros and 20 consecutive ones is generated. The length of this pattern is 1,048,575 bits.

- *prbs15* (Pseudo Random Bit Sequence-15)

PRBS15 is a PRBS pattern where a maximum of 14 consecutive zeros and 15 consecutive ones is generated. The length of this pattern is 32,767 bits.

PRBS20 and PRBS15 are not normally used to test T1 services because they violate the ones density and/or consecutive zeros restrictions for T1 signals. However, they may be used to simulate customer traffic for testing a channel (DS-0) in a Drop/Insert mode. Since they do not stress timing recovery, ALBO or equalizer circuits they are not as useful as the QRSS pattern.

<duration> can be configured for 10-300 seconds.

<loop-up> is the loop code to be sent to the far-end prior to running the BERT test for the Ds1.

- **noloop**

A loop up message is not sent on the data link before the BERT pattern is generated. The **noloop** parameter can be used if the customer wishes to use a test set at the far end to measure the BERT pattern generated by the MALC while also sending the same pattern to the MALC. This can be used to determine which pair of the T1 is receiving bit errors.

- **lineloop**

The `lineloop` parameter provides for a loopback of the timeslots. Normal T1 BERT testing is done with the *lineloop* loopback test.

- **payloadloop**

The `payloadloop` parameter provides for a loopback of the Data-Link. This is a special test mode for the T1 data-link, aka overhead or payload.

Once the **ds1bert start** command is run, you display the BER test results using the **ds1bert show** command.

```
ds1bert show <interface>
```

Running a BER test on T1 EFM

- 1 Change the **adminstatus** parameter to *testing* before running the BER test.

```
zSH> update if-translate 1-8-1-0/ds1
if-translate 1-4-1-0/ds1
Please provide the following: [q]uit.
ifIndex: -----> {1203}:
shelf: -----> {1}:
slot: -----> {8}:
port: -----> {1}:
subport: -----> {0}:
type: -----> {ds1}:
adminstatus: -----> {up}: testing
physical-flag: -----> {true}:
iftype-extension: --> {none}:
ifName: -----> {1-8-1-0}:
redundancy-param1: -> {0}:
description-index: -> {0}: ** read-only **
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

- 2 Run the BER test.

Run a BERT for T1 EFM by entering the **ds1bert start** command.

```
zSH> ds1bert start 1-8-1-0/ds1 qrss 70 lineloop
Bert started on interface 25, use "ds1bert show" cmd to get results.
```

- 3 View results of the BER test

```
zSH> ds1bert show 1-8-1-0/ds1
Bert Information:
-----
Bert Type.....qrss
Bert Duration.....70
Farend Loopback.....lineloop

***** Bert Statistics of Line 1633 *****

Bert Status: IN-PROGRESS
```

```

SEC      ES      OOS      ERR
-----
1         0         0         0
*****
                                End
*****

```

4 Restore operation of the T1 circuit by returning the **adminstatus** to *up*.

```

zSH> update if-translate 1-8-1-0/ds1
if-translate 1-4-1-0/ds1
Please provide the following: [q]uit.
ifIndex: -----> {1203}:
shelf: -----> {1}:
slot: -----> {8}:
port: -----> {1}:
subport: -----> {0}:
type: -----> {ds1}:
adminstatus: -----> {testing}: up
physical-flag: -----> {true}:
iftype-extension: --> {none}:
ifName: -----> {1-8-1-0}:
redundancy-param1: -> {0}:
description-index: -> {0}:  ** read-only **
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```


17

MXK T1/E1 PSEUDO WIRE EMULATION (PWE) CARD

This chapter describes the MXK-PWE-T1/E1-24 card. This chapter includes:

- [PWE T1/E1 24-port line card, page 1509](#)
- [T1/E1 24 port TDM cables, page 1514](#)

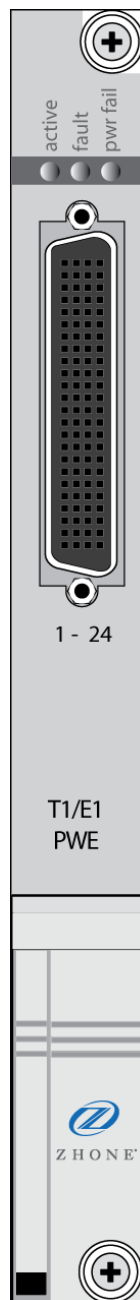
For information about configuring PWE on the MXK-PWE-T1/E1-24, please see [Chapter 7, MXK Pseudo Wire Emulation \(PWE\) Configuration](#), on page 549.

PWE T1/E1 24-port line card

This section describes the PWE T1/E1 line card and how to configure the card:

- [PWE T1/E1 24-port line card overview, page 1510](#)
- [PWE T1/E1 24-port line card specifications, page 1511](#)
- [PWE T1/E1 24-port line card configuration, page 1511](#)

PWE T1/E1 24-port line card overview



The MXK-PWE-T1/E1-24 line card allows T1/E1 circuits to be carried over a PSN with Zhone's implementation of Pseudo Wire Emulation (PWE).

PWE is a circuit emulation service (CES) which supports PWE3 Edge-To Edge Emulation (RFC 3985) over a packet switched network (PSN).

The MXK-PWE-T1/E1-24 line card supports reporting Alarm Indication Signal (AIS) alarms received from an attached device.

PWE T1/E1 24-port line card specifications

Table 220: MXK-PWE-T1/E1-24 port card specifications

Specification	Description
Size	1 slot
Density	24 ports T1/E1
Connectors	One (1) 96 pin Molex connector for 24 T1 or E1 circuits
Standards supported	IETF-PWE3 TDMoIP ITU-T G.823/824 ITU-T Y.1413, Y.1414 ITU-T Y.1452, 1453 D4 and ESF per T1.403 IEEE 802.3 Ethernet 802.1Q, 802.1p
Line encoding methods supported	AMI, B8ZS, HDB3
Supported line rates	1.544 MHz, 2.048 MHz
Power	18 Watts nominal 20 Watts maximum

PWE T1/E1 24-port line card configuration

Configuring a MXK-T1/E1 PWE card

The following example creates a **card-profile** for an T1/E1 PWE 24-port card in *shelf* 1, *slot* 7. Note that the **linetype** parameter must be selected to create the card profile.

```
zSH> card add 7 linetype ds1
```

or

```
zSH> new card-profile 1/7/10215
sw-file-name: -----> {mxlc24t1e1pwe.bin}:
admin-status: -----> {operational}:
upgrade-sw-file-name: ---> {}:
upgrade-vers: -----> {}:
admin-status-enable: ----> {enable}:
sw-upgrade-admin: -----> {reloadcurrrev}:
sw-enable: -----> {true}:
sw-upgrade-enable: -----> {false}:
card-group-id: -----> {0}:
hold-active: -----> {false}:
weight: -----> {nopreference}:
card-line-type: -----> {}:ds1
```

```

card-atm-configuration: -> {notapplicable}:
card-line-voltage: -----> {not-used}:
maxvpi-maxvci: -----> {notapplicable}:
card-init-string: -----> {}:
wetting-current: -----> {disabled}:
pwe-timing-mode: -----> {none}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s

```

Testing T1/E1

Configuring T1/E1 loopbacks

Before configuring a loopback on a ds1 line on the PWE blades you first must have a PWE bundle created. The act of creating the bundle initializes the hardware. If the PWE bundle is not created (and the hardware is not initialized), the loopback will not work

T1/E1 loopbacks are initiated by updating the associated **ds1-profile** and setting the **loopback-config** parameter to one of four (4) available options:

- *noloop* (default)
- *lineloop*
- *localloop*
- *payloadloop*

- 1** Before configuring a loopback on ds1 lines on PWE line cards, the PWE bundle must be created.

See [MXK Pseudo Wire Emulation \(PWE\) Configuration, page 549](#) for information about configuring PWE connections.

- 2** Before initiating T1/E1 loopbacks, you must first set the **adminstatus** of the line to the **testing** state.

```

zSH> update if-translate 1-8-1-0/ds1
if-translate 1-8-1-0/ds1
Please provide the following: [q]uit.
ifIndex: -----> {1633}:
shelf: -----> {1}:
slot: -----> {8}:
port: -----> {1}:
subport: -----> {0}:
type: -----> {ds1}:
adminstatus: -----> {up}: testing
physical-flag: -----> {true}:
iftype-extension: --> {none}:
ifName: -----> {1-8-1-0}:
redundancy-param1: -> {0}:
description-index: -> {0}: ** read-only **
.....

```

Save changes? [s]ave, [c]hange or [q]uit: **s**
Record updated.

3 Set loopback-config to the appropriate option (in this example, lineloop)

```
zSH> update ds1-profile 1-8-1-0/ds1
Please provide the following: [q]uit.
line-type: -----> {esf}:
line-code: -----> {b8zs}:
send-code: -----> {sendnocode}:
circuit-id: -----> {ds1}:
loopback-config: -----> {noloop}: lineloop
signal-mode: -----> {robbedbit}:
fdl: -----> {fdlnone}:
.
.
.
.....
```

Save changes? [s]ave, [c]hange or [q]uit: **s**

Record updated.

T1/E1 24 port TDM cables

Cabling options include the MXK-CBL-T1/E1-24-45DEG and the following blunt cables:

- MXK-CBL-T1/E1-24-15FT-BLUNT
- MXK-CBL-T1/E1-24-30FT-BLUNT
- MXK-CBL-T1/E1-24-100FT-BLUNT
- MXK-CBL-T1/E1-24-350FT-BLUNT

MXK-CBL-T1/E1-2-45DEG

Figure 178 shows the MXK EFM T1/E1 24-port bonding cable (MXK-CBL-T1/E1-24-45DEG and MALC-CBL-T1/E1-24-45DEG-20FT). Table 224 on page 1517 lists the pinouts.

Figure 178: MXK T1/E1 24 port cable

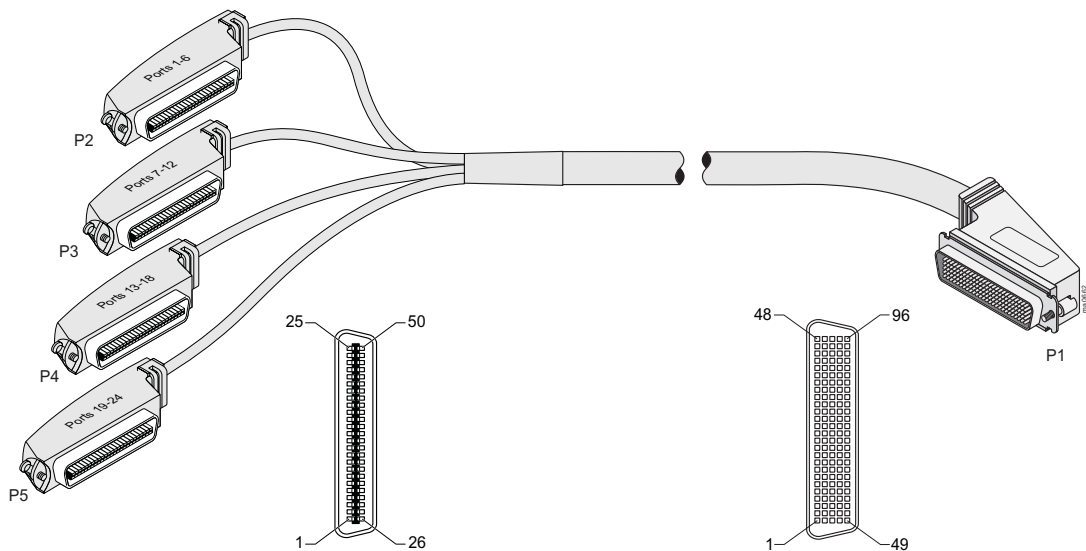


Table 221: Port-Pair Detail Ports 1-6 (P1 to P2) for MXK T1/E1 24 port cable

Port	Pair	Signal	Color	From	To
1	1	TX 1 Ring	BLU/WHT	P1-1	P2-1
		TX 1 Tip	WHT/BLU	P1-2	P2-26
2	2	RX 1 Ring	ORG/WHT	P1-3	P2-27
		RX 1 Tip	WHT/ORG	P1-4	P2-2
2	3	TX 2 Ring	GRN/WHT	P1-5	P2-5
		TX 2 Tip	WHT/GRN	P1-6	P2-30

Table 221: Port-Pair Detail Ports 1-6 (P1 to P2) for MXK T1/E1 24 port cable

Port	Pair	Signal	Color	From	To
	4	RX 2 Ring	BRN/WHT	P1-7	P2-31
		RX 2 Tip	WHT/BRN	P1-8	P2-6
3	5	TX 3 Ring	SLT/WHT	P1-9	P2-9
		TX 3 Tip	WHT/SLT	P1-10	P2-34
	6	RX 3 Ring	BLU/RED	P1-11	P2-35
		RX 3 Tip	RED/BLU	P1-12	P2-10
4	7	TX 4 Ring	ORG/RED	P1-13	P2-13
		TX 4 Tip	RED/ORG	P1-14	P2-38
	8	RX 4 Ring	GRN/RED	P1-15	P2-39
		RX 4 Tip	RED/GRN	P1-16	P2-14
5	9	TX 5 Ring	BRN/RED	P1-17	P2-17
		TX 5 Tip	RED/BRN	P1-18	P2-42
	10	RX 5 Ring	SLT/RED	P1-19	P2-43
		RX 5 TIP	RED/SLT	P1-20	P2-18
6	11	TX 6 Ring	BLU/BLK	P1-21	P2-21
		TX 6 Tip	BLK/BLU	P1-22	P2-46
	12	RX 6 Ring	ORG/BLK	P1-23	P2-47
		RX 6 TIP	BLK/ORG	P1-24	P2-22

Table 222: Port-Pair Detail Ports (P1 to P3) 7-12 for MXK T1/E1 24 port cable

Port	Pair	Signal	Color	From	To
7	13	TX 7 Ring	BLU/WHT	P1-25	P3-1
		TX 7 Tip	WHT/BLU	P1-26	P3-26
	14	RX 7 Ring	ORG/WHT	P1-27	P3-27
		RX 7 Tip	WHT/ORG	P1-28	P3-2
8	15	TX 8 Ring	GRN/WHT	P1-29	P3-5
		TX 8 Tip	WHT/GRN	P1-30	P3-30
	16	RX 8 Ring	BRN/WHT	P1-31	P3-31
		RX 8 Tip	WHT/BRN	P1-32	P3-6

Table 222: Port-Pair Detail Ports (P1 to P3) 7-12 for MXK T1/E1 24 port cable

Port	Pair	Signal	Color	From	To
9	17	TX 9 Ring	SLT/WHT	P1-33	P3-9
		TX 9 Tip	WHT/SLT	P1-34	P3-34
	18	RX 9 Ring	BLU/RED	P1-35	P3-35
		RX 9 Tip	RED/BLU	P1-36	P3-10
10	19	TX 10 Ring	ORG/RED	P1-37	P3-13
		TX 10 Tip	RED/ORG	P1-38	P3-38
	20	RX 10 Ring	GRN/RED	P1-39	P3-39
		RX 10 Tip	RED/GRN	P1-40	P3-14
11	21	TX 11 Ring	BRN/RED	P1-41	P3-17
		TX 11 Tip	RED/BRN	P1-42	P3-42
	22	RX 11 Ring	SLT/RED	P1-43	P3-43
		RX 11 Tip	RED/SLT	P1-44	P3-18
12	23	TX 12 Ring	BLU/BLK	P1-45	P3-21
		TX 12 Tip	BLK/BLU	P1-46	P3-46
	24	RX 12 Ring	ORG/BLK	P1-47	P3-47
		RX 12 Tip	BLK/ORG	P1-48	P3-22

Table 223: Port-Pair Detail Ports (P1 to P4)13-18 for MXK T1/E1 24 port cable

Port	Pair	Signal	Color	From	To
13	25	TX 13 Ring	BLU/WHT	P1-49	P4-1
		TX 13 Tip	WHT/BLU	P1-50	P4-26
	26	RX 13 Ring	ORG/WHT	P1-51	P4-27
		RX 13 Tip	WHT/ORG	P1-52	P4-2
14	27	TX 14 Ring	GRN/WHT	P1-53	P4-5
		TX 14 Tip	WHT/GRN	P1-54	P4-30
	28	RX 14 Ring	BRN/WHT	P1-55	P4-31
		RX 14 Tip	WHT/BRN	P1-56	P4-6

Table 223: Port-Pair Detail Ports (P1 to P4)13-18 for MXK T1/E1 24 port cable

Port	Pair	Signal	Color	From	To
15	29	TX 15 Ring	SLT/WHT	P1-57	P4-9
		TX 15 Tip	WHT/SLT	P1-58	P4-34
	30	RX 15 Ring	BLU/RED	P1-59	P4-35
		RX 15 Tip	RED/BLU	P1-60	P4-10
16	31	TX 16 Ring	ORG/RED	P1-61	P4-13
		TX 16 Tip	RED/ORG	P1-62	P4-38
	32	RX 16 Ring	GRN/RED	P1-63	P4-39
		RX 16 Tip	RED/GRN	P1-64	P4-14
17	33	TX 17 Ring	BRN/RED	P1-65	P4-17
		TX 17 Tip	RED/BRN	P1-66	P4-42
	34	RX 17 Ring	SLT/RED	P1-67	P4-43
		RX 17 Tip	RED/SLT	P1-68	P4-18
18	35	TX 18 Ring	BLU/BLK	P1-69	P4-21
		TX 18 Tip	BLK/BLU	P1-70	P4-46
	36	RX 18 Ring	ORG/BLK	P1-71	P4-47
		RX 18 Tip	BLK/ORG	P1-72	P4-22

Table 224: Port-Pair Detail Ports (P1 to P5) 19-24 for MXK T1/E1 24 port cable

Port	Pair	Signal	Color	From	To
19	37	TX 19 Ring	BLU/WHT	P1-73	P5-1
		TX 19 Tip	WHT/BLU	P1-74	P5-26
	38	RX 19 Ring	ORG/WHT	P1-75	P5-27
		RX 19 Tip	WHT/ORG	P1-76	P5-2
20	39	TX 20 Ring	GRN/WHT	P1-77	P5-5
		TX 20 Tip	WHT/GRN	P1-78	P5-30
	40	RX 20 Ring	BRN/WHT	P1-79	P5-31
		RX 20 Tip	WHT/BRN	P1-80	P5-6

Table 224: Port-Pair Detail Ports (P1 to P5) 19-24 for MXK T1/E1 24 port cable

Port	Pair	Signal	Color	From	To
21	41	TX 21 Ring	SLT/WHT	P1-81	P5-9
		TX 21 Tip	WHT/SLT	P1-82	P5-34
	42	RX 21 Ring	BLU/RED	P1-83	P5-35
		RX 21 Tip	RED/BLU	P1-84	P5-10
22	43	TX 22 Ring	ORG/RED	P1-85	P5-13
		TX 22 Tip	RED/ORG	P1-86	P5-38
	44	RX 22 Ring	GRN/RED	P1-87	P5-39
		RX 22 Tip	RED/GRN	P1-88	P5-14
23	45	TX 23 Ring	BRN/RED	P1-89	P5-17
		TX 23 Tip	RED/BRN	P1-90	P5-42
	46	RX 23 Ring	SLT/RED	P1-91	P5-43
		RX 23 Tip	RED/SLT	P1-92	P5-18
24	47	TX 24 Ring	BLU/BLK	P1-93	P5-21
		TX 24 Tip	BLK/BLU	P1-94	P5-46
	48	RX 24 Ring	ORG/BLK	P1-95	P5-47
		RX 24 Tip	BLK/ORG	P1-96	P5-22

T1/E1 24 blunt cables

Several blunt-end T1/E1 24 cable options are supported. Note that the 24 port cable options use the same connector as 48 port ADSL options.

- MXK-CBL-T1/E1-24-15FT-BLUNT
- MXK-CBL-T1/E1-24-30FT-BLUNT
- MXK-CBL-T1/E1-24-100FT-BLUNT
- MXK-CBL-T1/E1-24-350FT-BLUNT

The following tables list the blunt cable pinouts.

Table 225: Pinout for 24 port T1/E1 to blunt end cable

Port	Pair	Signal	Color	Form	Binder Group
1	1	TX 1 Ring	Blue/White	P1-1	1 (Blue)
		TX 1 Tip	White/Blue	P1-2	
	2	RX 1 Ring	Orange/White	P1-3	
		TX 1 Tip	White/Orange	P1-4	
2	3	TX 2 Ring	Green/White	P1-5	
		TX 2 Tip	White/Green	P1-6	
	4	RX 2 Ring	Brown/White	P1-7	
		TX 2 Tip	White/Brow	n P1-8	
3	5	TX 3 Ring	Slate/White	P1-9	
		TX 3 Tip	White/Slate	P1-10	
	6	RX 3 Ring	Blue/Red	P1-11	
		TX 3 Tip	Red/Blue	P1-12	
4	7	TX 4 Ring	Orange/Red	P1-13	
		TX 4 Tip	Red/Orange	P1-14	
	8	RX 4 Ring	Green/Red	P1-15	
		TX 4 Tip	Red/Green	P1-16	
5	9	TX 5 Ring	Brown/Red	P1-17	
		TX 5 Tip	Red/Brown	P1-18	
	10	RX 5 Ring	Slate/Red	P1-19	
		TX 5 Tip	Red/Slate	P1-20	
6	11	TX 6 Ring	Blue/Black	P1-21	
		TX 6 Tip	Black/Blue	P1-22	
	12	RX 6 Ring	Orange/Black	P1-23	
		TX 6 Tip	Black/Orange	P1-24	

Table 226: Pinout for 24 port T1/E1 to blunt end cable (Cont'd)

Port	Pair	Signal	Color	Form	Binder Group
7	13	TX 7 Ring	Blue/White	P1-25	2 (Orange)
		TX 7 Tip	White/Blue	P1-26	
	14	RX 7 Ring	Orange/White	P1-27	
		TX 7 Tip	White/Orange	P1-28	
8	15	TX 8 Ring	Green/White	P1-29	
		TX 8 Tip	White/Green	P1-30	
	16	RX 8 Ring	Brown/White	P1-31	
		TX 8 Tip	White/Brown	P1-32	
9	17	TX 9 Ring	Slate/White	P1-33	
		TX 9 Tip	White/Slate	P1-34	
	18	RX 9 Ring	Blue/Red	P1-35	
		TX 9 Tip	Red/Blue	P1-36	
10	19	TX 10 Ring	Orange/Red	P1-37	
		TX 10 Tip	Red/Orange	P1-38	
	20	RX 10 Ring	Green/Red	P1-39	
		TX 10 Tip	Red/Green	P1-40	
11	21	TX 11 Ring	Brown/Red	P1-41	
		TX 11 Tip	Red/Brown	P1-42	
	22	RX 11 Ring	Slate/Red	P1-43	
		TX 11 Tip	Red/Slate	P1-44	
12	23	TX 12 Ring	gBlue/Black	P1-45	
		TX 12 Tip	Black/Blue	P1-46	
	24	RX 12 Ring	Orange/Black	P1-47	
		TX 12 Tip	Black/Orange	P1-48	

Table 227: Pinout for 24 port T1/E1 to blunt end cable (Cont'd)

Port	Pair	Signal	Color	Form	Binder Group
13	25	TX 13 Ring	Blue/White	P1-49	3 (Green)
		TX 13 Tip	White/Blue	P1-50	
	26	RX 13 Ring	Orange/White	P1-53	
		TX 13 Tip	White/Orange	P1-52	
14	27	TX 14 Ring	Green/White	P1-55	
		TX 14 Tip	White/Green	P1-54	
	28	RX 14 Ring	Brown/White	P1-57	
		TX 14 Tip	White/Brown	P1-56	
15	29	TX 15 Ring	Slate/White	P1-59	
		TX 15 Tip	White/Slate	P1-58	
	30	RX 15 Ring	Blue/Red	P1-61	
		TX 15 Tip	Red/Blue	P1-60	
16	31	TX 16 Ring	Orange/Red	P1-63	
		TX 16 Tip	Red/Orange	P1-62	
	32	RX 16 Ring	Green/Red	P1-65	
		TX 16 Tip	Red/Green	P1-64	
17	33	TX 17 Ring	Brown/Red	P1-67	
		TX 17 Tip	Red/Brown	P1-66	
	34	RX 17 Ring	Slate/Red	P1-69	
		TX 17 Tip	Red/Slate	P1-68	
18	35	TX 18 Ring	Blue/Black	P1-71	
		TX 18 Tip	Black/Blue	P1-70	
	36	RX 18 Ring	Orange/Black	P1-73	
		TX 18 Tip	Black/Orange	P1-72	

Table 228: Pinout for 24 port T1/E1 to blunt end cable (Cont'd)

Port	Pair	Signal	Color	Form	Binder Group
19	37	TX 19 Ring	Blue/White	P1-73	4 (Brown)
		TX 19 Tip	White/Blue	P1-74	
	38	RX 19 Ring	Orange/White	P1-75	
		TX 19 Tip	White/Orange	P1-76	
20	39	TX 20 Ring	Green/White	P1-77	
		TX 20 Tip	White/Green	P1-78	
	40	RX 20 Ring	Brown/White	P1-79	
		TX 20 Tip	White/Brown	P1-80	
21	41	TX 21 Ring	Slate/White	P1-81	
		TX 21 Tip	White/Slate	P1-82	
	42	RX 21 Ring	Blue/Red	P1-83	
		TX 21 Tip	Red/Blue	P1-84	
22	43	TX 22 Ring	Orange/Red	P1-85	
		TX 22 Tip	Red/Orange	P1-86	
	44	RX 22 Ring	Green/Red	P1-87	
		TX 22 Tip	Red/Green	P1-88	
23	45	TX 23 Ring	Brown/Red	P1-89	
		TX 23 Tip	Red/Brown	P1-90	
	46	RX 23 Ring	Slate/Red	P1-91	
		TX 23 Tip	Red/Slate	P1-92	
24	47	TX 24 Ring	Blue/Black	P1-93	
		TX 24 Tip	Black/Blue	P1-94	
	48	RX 24 Ring	Orange/Black	P1-95	
		TX 24 Tip	Black/Orange	P1-96	

18

MXK TEST ACCESS CARDS

This chapter describes the MXK Test Access (TAC) cards and explains how to configure it. The chapter includes:

- [TAC cards, page 1523](#)
- [Configure TAC cards, page 1530](#)
- [Performing line test using TAC cards with external testing set, page 1533](#)
- [Performing internal line test with TAC-ITM-RING card, page 1537](#)
- [Configuring external alarms, page 1562](#)
- [Configuring an external clock, page 1562](#)
- [Connecting an external ring source, page 1565](#)
- [TAC cards pinouts, page 1568](#)

TAC cards

This section describes the MXK TAC cards and how to use them including:

- [TAC card overview, page 1524](#)
- [TAC card specifications, page 1525](#)
- [Connectors on the TAC cards, page 1526](#)
- [Metallic loop testing, page 1527](#)
- [Internal look out line test, page 1527](#)
- [Ring generator, page 1528](#)

TAC card overview



MXK-TAC-ITM-RING card is a single slot card that supports metallic loop testing for DSL and POTS interfaces with or without the external test set. The metallic loop testing without external test set is also called look-out internal line testing. Note that the type of tests provided will vary, depending on the type of card being tested

It also supports external alarm inputs (12 circuits, wet or dry, normally open or normally closed), T1/E1 or BITS external network clock access, and ring generation (internal ring generator or access for an external ring generator)

The MXK TAC cards provide metallic test access to verify the local loop conditions, perform line testing on distant regions of the physical copper cable connecting the MXK and remote devices. It can assess breakages in the cable, identifying the following data:

- **Distance.** Identifies the amount of distance between the TAC card and the location of the break or open that occurred on the copper cable.
- **Shorts.** Identifies the port to which a cable containing an electrical short is connected.
- **Unbalance.** Identifies if one side is longer between the tip and the ring, creating an unbalance in the connection.

- **Metallic noise.** Identifies any impairments on the cable that indicate an interruption on the ring.



Note: The MXK supports only one active TAC card at a time and a total of two TAC cards in the system.

TAC card specifications

Table 229: TAC card specifications

Specification	Value
Size	1 slot
Physical interfaces	• Metallic test access port: An RJ45 connector that connects to the external test set. It connects the external test set to metallic test bus on backplane (supports one port test simultaneously in system). • External test set control port: A serial control RS232D signalling port on RJ45 connector that provides a control connection to the external test set. • External clock input port: An RJ45 connector that accepts T1/E1 or BITS external clock reference (all versions), provisionable as system clock source. • External ring generator input port: A two position plug spaced at 5.08mm conforming to the IEC 60664-1 industry standard, such as the RIA Type 249 part number 312491 02. This port connects to the external ring generator. • External alarm connector: A 26 pin D sub connector that supports 12 alarm closures for detecting various alarm types from collocated equipment. Supports isolated closure, ground and –48VDC closure (states and names provisionable in software).
Metallic test functions	Look-out testing (toward the loop) for ADSL, ULC, and POTS interfaces (with the exception of ADSL 32 cards).  Note: The type of tests provided will vary, depending on the type of card being tested.

Table 229: TAC card specifications (Continued)

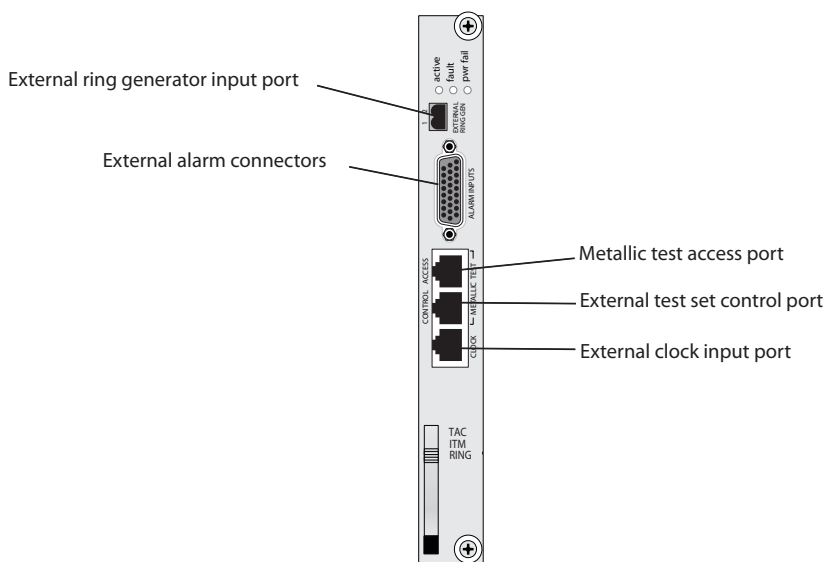
Specification	Value
Ring generation	External ring generator voltage connector. Internal ring voltage sine wave generator: 34 REN @ 93VRMS for the TAC cards
Redundancy	1+1 card redundancy
Clocking	TAC cards can be configured to use T1, E1, or 2 MHz signal as the clock source. The clocking reference on the TAC card with 2 MHz BITS clock complies with ITU-T G.703 standard.
Accuracy field	-10% to +10%
Power consumption	13 Watts nominal (no ringer load), 53 Watts maximum at full ringing load for the TAC cards.

Connectors on the TAC cards

TAC-ITM-RING cards have following connectors:

- Metallic test access port
- External test set control port
- External clock input port
- External ring generator input port
- External alarm connectors

[Figure 179](#) shows the connectors on the TAC-ITM-RING card.

Figure 179: Connectors on TAC-ITM-RING card

Metallic loop testing

The TAC cards support metallic loop testing for T1, POTS, and DSL loops, providing preventive measures for potential line breaks.

The TAC cards not only support external test sets, they also provide internal look-out line testing. External test sets supported include Tollgrade, Harris/Fluke, and Teradyne 4-Tel components.

Internal look out line test

Internal line testing is supported by the TAC card. With its own integrated test set, the TAC card in each shelf can perform test out session without the external test set.

Some line cards have the integrated line testing functionality, they don't need the TAC card, such as the MXK-POTS-72, MXK-VDSL2-POTS-BCM-17A-24, and the MXK-ADSL2+-POTS-BCM-48A-RNG-2S (with card line-type=adsl-pots-pv-rng-itm).



Note: While adding an MXK-ADSL2+-POTS-BCM-48A-RNG-2S card with the card line type **adsl-pots-pv-rng-itm**, it has enabled the integrated ITM test functionality on the card, access to the test functionality of the TAC cards is blocked. If you want to use the TAC card to do the line testing, add the MXK-ADSL2+-POTS-BCM-48A-RNG-2S card with card line type **adsl-pots-pv**.

Cards supporting look-out test access

The TAC cards provide access to external test equipment through an RJ45 connector for look-out test access. All ADSL2-48 cards and EFM cards support look-out test access.

The following table provides examples of common instances of these card types that support internal test relay to TAC cards and Look-out test access to the external test equipment.

Table 230: Examples of common cards supporting internal test relay and look-out test access

Card Type	Example
ADSL2-48	MXK-ADSL2+BCM-48A
	MXK-ADSL2+BCM-48B
	MXK-ADSL2+SPLTR600-BCM-48A-2S
	MXK-ADSL2+SPLTR900-BCM-48A-2S
	MXK-ADSL2+-POTS-BCM-48A-2S
	MXK-ADSL2+-POTS-BCM-48A-RNG-2S (with line type adsl-pots-pv)
EFM	MXK-EFM-SHDSL-24-NTWC
	MXK-EFM-SHDSL-24-NTP
	MXK-EFM-T1E1-24

Ring generator

The TAC cards contain the ring generator for MXK-ADSL2+-POTS-BCM-48A-2S card installed in the MXK. Ringing voltage is supplied to this installed ADSL POTS combo card via a backplane bus. Note that only one TAC card can supply ringing voltage to the system at a time.

Some POTS cards have their own ring generator on the cards, such as MXK-ADSL2+-POTS-BCM-48A-RNG-2S and POTS 72 card.

The TAC cards also contain a ringing voltage detector that senses the absence or faulty of ringing voltage on the card itself, or on an external ringing generator (if one exists).

- If the ringing voltage detector detects the absence of ringing voltage, a “Ringer source not detected” error message will be generated. The redundant TAC card can supply the ringing voltage, or the MXK can be configured to use another external ringing generator.



Note: The MXK ground wires must be tied to the +48V battery return at the main power Distribution Center. Absence of this connection can cause malfunctions on some cards, including generation of the TAC card error message “Ringer source not detected”.

- If the ringing voltage exceeds the limit, the ringer voltage will be turned off, and a “Internal ringer disabled” error message will be generated. Software will attempt to restart it every 1 second. When the ring load drops back to normal, the TAC internal ringer will automatically recover after 2 seconds, and a message “Ringer source detected” will be generated.

Configure TAC cards



Caution: Each TAC card in a redundant pair must be configured identically; the cards do not share state or configuration information. In addition, the user must manually keep the configuration of the active and standby cards in sync. This applies to both a matched pair and a mixed pair of TAC cards.

Each TAC card installed in the system must have a **card-profile**. Each type of slot card requires different settings in the **card-profile**.

TAC cards have the following types and software images:

Table 231: TAC card types

Card	Type	Name of software image
MXK-TAC-ITM-RING	5072	<i>tacitmring.bin</i>

Creating card profiles for TAC cards

Creating card profiles for TAC-ITM-RING cards

The **card-profile** for TAC-ITM-RING cards require that the **card-line-type** (which specifies the external clock source type) be specified.

To configure a redundant TAC-ITM-RING card, create a second **card-profile** for the redundant card.

To enable a TAC-ITM-RING card:

```
zSH> card add 16 linetype ds1 group 2
An autogenerated card-group-id [2] is assigned for this card type.
new card-profile 1/16/5072 added, sw-file-name "tacitmring.bin", 2 options:
card-group-id 2  card-line-type ds1
```

or

```
zSH> new card-profile 16
Please provide the following: [q]uit.
sw-file-name: -----> {tacitmring.bin}:
admin-status: -----> {operational}:
upgrade-sw-file-name: ---> {}:
upgrade-vers: -----> {}:
admin-status-enable: ----> {enable}:
sw-upgrade-admin: -----> {reloadcurrrev}:
sw-enable: -----> {true}:
sw-upgrade-enable: -----> {false}:
card-group-id: -----> {0}:2
hold-active: -----> {false}:
weight: -----> {nopreference}:
card-line-type: -----> {unknowntype}: ds1
card-atm-configuration: -> {notapplicable}:
card-line-voltage: -----> {not-used}:
```

```

maxvpi-maxvci:-----> {notapplicable}:
card-init-string:-----> {}:
wetting-current:-----> {disabled}:
pwe-timing-mode:-----> {none}:
.....
Save new record? [s]ave, [c]hange or [q]uit:  s
An autogenerated card-group-id [2] is assigned for this card type.
New record saved.

```

Verifying the slot card installation

After saving the **card-profile** record, the slot card in that slot resets and begins downloading their software image from the flash card. This could take a few moments.

When the card has finished loading, a log message similar to the following is displayed (if logging is enabled):

```
zSH> Card in slot slot-number changed state to RUNNING
```

You can also use the **slots** command and specify the slot number of the card to view the state of the card. For example:

```

zSH> slots 16
Type                : *TAC ITM RING
Card Version        : 800-01762-01-N
EEPROM Version      : 1
Serial #            : 1763389
CLEI Code           : No CLEI
Card-Profile ID     : 1/16/5072
Shelf               : 1
Slot                : 16
ROM Version         : MALC CAN 1.13.0.108
Software Version    : MXK 1.16.2.028
State               : LOADING indicates the card is still initializing
Mode                : FUNCTIONAL
Heartbeat check     : enabled
Longest hbeat       : 0
Fault reset         : enabled
Uptime              : 13 minutes

```

To view the status of all the cards, use the **slots** command without any arguments:

```

zSH> slots
Uplinks
  a: *MXK EIGHT GIGE (RUNNING)
  b: MXK EIGHT GIGE (RUNNING)
Cards
  1: MXK 8 PORT GPON (RUNNING)
 16: *TAC ITM RING (RUNNING)

```

An asterisk (*) indicates the card is an active card.

Viewing active redundant cards

You can also use the **showactivecards** command to view all active cards in the system that are part of a redundant card group:

```
zSH> showactivecards
Shelf/Slot  Group Id  Card Type
-----
1:  1/16  2          TAC ITM RING
2:  1/a   1          MXK EIGHT GIGE
```


Performing line test using TAC cards with external testing set

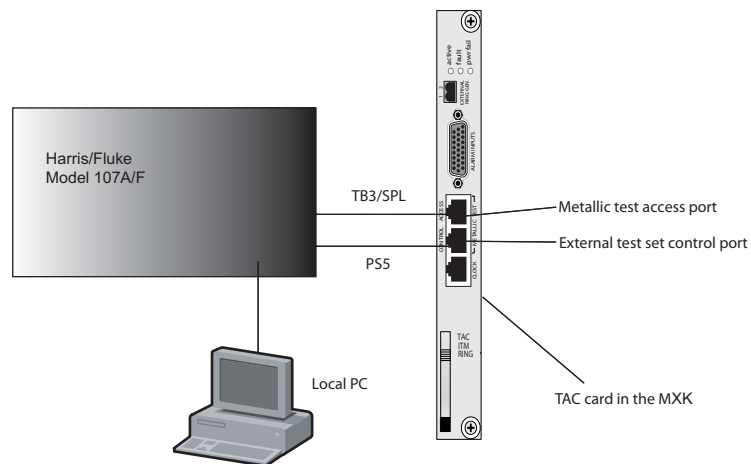
The TAC family of cards support external line testing.

Connecting the external test set to TAC card

The external test set is connected to the TAC card through the metallic test access port and the external test set control port. The following figure details how an external test set can be connected to the TAC card. (external test sets are also known as external test heads, external test units, and remote test units.)

The MXK enables connecting the external test set to the TAC card to set test relays. The default baud rate is 9600 bps. (This can be changed by modifying the **rs232-profile**.)

Figure 180: External test set connected to a TAC card



For example, to test the integrity of a line by Harris external test set, issue the **test aid** command, using the shelf, slot, and port, as a numeric keyword. For shelf 1/slot 5/port 1, issue the command **test aid=1-5-1**. Sample output is provided below.

```
HARRIS>test aid=1-5-1
DN: PAIR: SITE: TEST CHAN: 07/18/2006 15:00
NLT: PASS LDT: N/A NPA: 910 CO: CLLI:OKLAND
AID: 1-5-1 ACC:TRUNK-WB COND: OUTWARD TTYPE: LOOP SUFF:
DC SIGNATURE AC SIGNATURE NOISE
KOHMS VOLTS KOHMS VOLTS CPE CAP 60HzINDUCED
C-MESSAGEdBrcC
9999 0.00 9999 0.00 NO 0.00 T-R 37.5 TO GROUND
9999 0.00 9999 0.00 NO 0.00 T-G .002 mA T-g 0.00 METALLIC
9999 0.00 9999 0.00 NO 0.00 R-G .002 mA R-G NOISE BAL
```

```
0.00 Mutual () NOISE
UNBALANCE: 0.00% TIP LENGTH: .001 KF HIST VER: K UP, K DN
+-----++ +--+
| DLC |M| |M| CABLE +--+ +--+
| |a|=|D|=====|DP|====|CPE|
|DSLAM|T| |F| +--+ +--+
+-----++ +--+
VER35: OPEN IN EQUIPMENT
Dispatch: OFFICE (No CPE Seen)
```

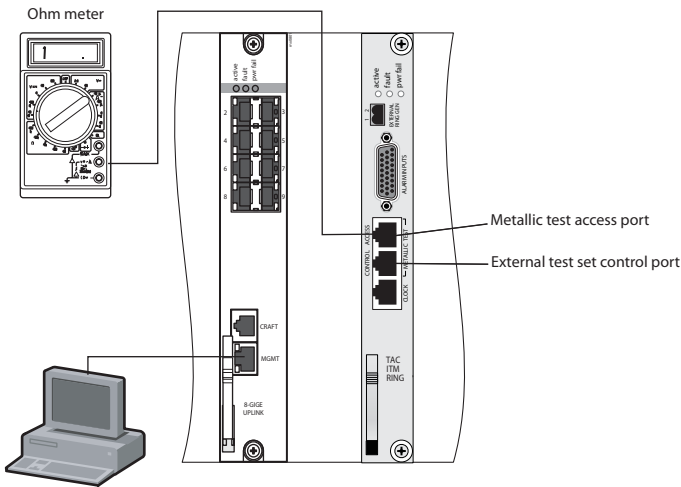
 **Note:** Refer to various external test set user guides for detail.

 **Note:** Only the pair of **Test out tip 1** and **Test out ring 1** is available to be used for loop testing.

Connecting the test measurement device to the metallic test access port

If the user wants to manually measure the line integrity, the user can connect the metallic test access port on the TAC card with a manual test measurement device, such as Ohm meter or voltage meter.

Figure 181: Manual test measurement device connected to a TAC card



After connecting the manual test measurement device, use the **mtac-linetest** command to set the relay options.

The following example enables a manual test measurement device to access to the ADSL interface on *shelf 1, slot 3, port 1*:

```
zSH> mtac-linetest 1/3/1 lookout none ds1
Successful - Test In Progress
```

To stop access to the interface, set the interface back to the defaults:

```
zSH> mtac-linetest 1/3/1 release none ds1
Mode is release, setting test id to none
```

The following example enables a manual test measurement device to access to the POTS interface on *shelf 1, slot 13, port 1*:

```
zSH> mtac-linetest 1/13/1 lookout none
Successful - Test In Progress
```

To stop access to the interface, set the interface back to the defaults:

```
zSH> mtac-linetest 1/13/1 release none
Mode is release, setting test id to none
```

To perform a look-out test on a T1/E1 circuit MXK T1/E1 line card, specify **ds1** as the line type:

```
zSH> mtac-linetest 1/3/1 lookout none ds1
Successful - Test In Progress
```

To stop access to the T1/E1 interface, set the interface back to the defaults:

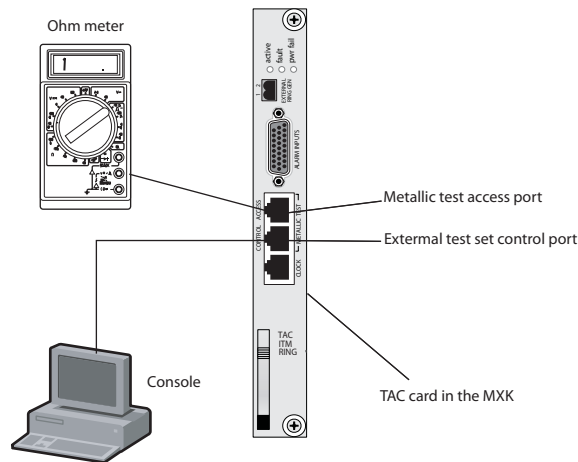
```
zSH> mtac-linetest 1/3/1 release none ds1
Mode is release, setting test id to none
```



Note: The interface must be set back to its defaults before a line can be specified for test access.

Connecting a console to the external test set control port

The user also can connect the external test set control port on the TAC card with a console to input commands. The metallic test access port on the TAC card would be connected with a manual test measurement device, such as Ohm meter or voltage meter to read the test results.

Figure 182: Console connected to a TAC card

Note: These commands are used on the TAC card external test set control port, not on the MXK uplink card zhone shell.

Use the TAC external test set control port command to determine what the state of the card is, either in Idle or Test mode, and to determine whether the line test has been successful.

The TAC external test set control port command is:

```
> mtac-linetest portaddr mode [linetype] [force]
```

Note that the force parameter can only performs on voicefxs lines.

```
> mtac-linetest 1/13/1 lookout
Successful - In TestMode
> mtac-linetest 1/13/1 release
Successful - Returned to operational state
> mtac-linetest 1/13/1 lookout adsl
Successful - In TestMode
> mtac-linetest 1/13/1 release
Successful - Returned to operational state
```

Performing internal line test with TAC-ITM-RING card

The TAC-ITM-RING card is able to perform Metallic Loop Testing (MLT) without an external test set.

The TAC-ITM-RING card comes with an integrated test head, thus, each shelf has its own dedicated test head. This means user can now communicate with each shelf concurrently to perform test out sessions.

This section describes the following information:

- [Working with the TAC line test command on page 1537](#)
- [Test IDs on page 1539](#)
- [Metallic loop tests on page 1541](#)
- [Troubleshooting with metallic loop tests on page 1557](#)
- [Auto-calibration on page 1560](#)
- [Lookout block diagram on page 1560](#)

Working with the TAC line test command

Use **mtac-linetest** commands to specify the test mode (lookout) and other test parameters for the internal line test.

The **mtac-linetest** command syntax is:

```
mtac-linetest portaddr mode testid [linetype] [force]
```

The **mtac-linetest** command has the required components of port address, mode, and test identifier; the optional components of linetype and parameter force.

```
zSH> mtac-linetest
Valid options for testid:
none abort foreigndcvoltage foreignacvoltage
dcloopresistance 3elementresistance 3elementcapacitance
receiveroffhook distancetoopen foreignaccurrents
ringerequiv dtmfandpulsedigitmeasurement noisemeasurement
tonegeneration transhybridloss drawandbreakdialtone
dcfeedselftest onandoffhookmeasurement ringingselftest
meteringselftest transmissionselftest howlertest
readloopandbatteryconditions
Valid parameters for testid:
    tonegeneration: p1=duration[sec] p2=freq[hz]
Usage: mtac-linetest <portaddr> <mode> <testid>
[<linetype>] [force] [p1] [p2] [p3] [p4] [p5]
Description:
    Execute tac test
Arguments:
    <portaddr> - port address in shelf/slot/port
    <mode> - lookout, lookin, release, bridge
    <testid> - A builtin tac linetest
```

<linetype> - adsl, ds1, shdsl, isdn, vdsl, voiceebs and voicefxs. Default voicefxs.

force - override option regardless if line is in use

p(1-5) - optional parameters for individual line test

[Table 232](#) lists supported parameters in the mtac-linetest command.

Table 232: mtac-linetest command parameters description

Parameter	Description
portaddr	Specifies the port address of the physical line to be tested. Values: A port address on the system. In the format <i>shelf/slot/port</i>
mode	Specifies metallic test mode for a given line. The test mode can be changed only if the port address parameter is set to a nonzero value. Values: Lookout The MXK service port is disconnected and the subscriber line is metalically routed to the TAC metallic test access port. This allows the testing of line with or without a subscriber terminal. Release Terminate the TAC test that in progress. Lookin and Bridge are not supported in current version. Default: Release
testid	Specifies the supported line tests. Values: none, abort, foreigndcvoltage, foreignacvoltage, dcloopresistance, 3elementresistance, 3elementcapacitance, receiveroffhook, distancetoopen, foreignaccurrents, ringerequiv, dtmfandpulsedigitmeasurement, noisemeasurement, tonegeneration, transhybridloss, drawandbreakdialtone, dcfeedselftest, onandoffhookmeasurement, ringingselftest, ringingmonitor, meteringselftest, transmissionselftest, howlertest, readloopandbatteryconditions Refer to Table 233 on page 1539 for the detail description for each value.

Table 232: mtac-linetest command parameters description

Parameter	Description
linetype	Specifies the line connect type to an equipment port class. This parameter is optional. Values: adsl ds1 shdsl isdn vdsl voiceebs voicefxs Default: By default, the line type is voicefxs for POTS loops, and should be changed to the correct type when testing a loop other than POTS. Note that, the line type is voicefxs for Combo lines, not adsl.
force	This option allows seizure of a line that may be in use. Using the embedded testing is invasive to the line and should not be used for a line in use. If a line is in use and must be tested, the force option will override the current usage. Values: force

Test IDs

[Table 233](#) lists the detailed description of the internal line tests that supported by TAC-ITM-RING card.

Table 233: TAC-ITM-RING Internal Line Tests

Test ID	Description
3elementcapacitance	This test measures tip-to-ground (T-G), ring-to-ground (R-G), and tip-to-ring (T-R) capacitance and impedance.
3elementresistance	This test measures tip-to-ground (T-G), ring-to-ground (R-G), and tip-to-ring (T-R) resistance.
abort	Terminate the running test.
dcfeedselftest	This procedure verifies that the test hardware can drive currents into a load and measure the voltage across a load.
dcloopresistance	This test measures DC loop resistance using one of the following algorithms: Forward/Reverse Polarity or Offset Compensation.
distancetoopen	This test estimates the distance to an open-circuit by analyzing the results of a 3 elements resistance test and a 3 elements capacitance test.

Table 233: TAC-ITM-RING Internal Line Tests (Continued)

Test ID	Description
drawandbreakdialtone	This test verifies the capability of the line circuit to detect off-hook and on-hook, the communication channel to the switching center, and the voice path from the switching center. This test is performed with the call-processing function enabled on the line under test. Note that this test will be supported in the future release.
dtmfandpulsedigit measurement	This test detects and measures a DTMF digit, pulse digit, or hook-switch flash. Only one digit or flash is reported for each invocation of this test. By default, a single tone is output on the line during this test.
foreignaccurrents	This test measures foreign AC currents.
foreigndcvoltage	This test examines the loop for the existence of DC voltage leaking into a line from an external source.
foreignacvoltage	The foreign AC voltage test is examining the loop for the existence of AC voltage leaking onto a line from an external source.
howlertest	This procedure generates a Howler (Receiver Off-Hook) tone until the phone goes on-hook or a timeout condition is detected.
meteringselftest	This procedure verifies that the line card can generate a metering pulse. It drives a metering signal into both a resistive load and an open-circuit using the current Metering Profile applied to the line.
none	If used with lookout mode, will enable the relay tests with the TAC card. If used with release mode, then it restores the normal setting.
nosiemeasurement	This procedure performs an active or passive noise test. Various filters may be applied to the received signal during this test. The application can apply special AC transmission coefficients during this test if desired.
onandoffhook measurement	This procedure verifies that the line circuit can detect on-hook and off-hook events.
readloopandbattery conditions	This procedure measures the instantaneous loop resistance, loop currents, and loop and battery voltages. No filtering is done during the measurement, so the results may fluctuate from one reading to the next in the presence of AC induction on the line.
receiveroffhook	This test determines whether the receiver is off-hook by running the DC Loop Resistance Test twice with different test currents and analyzing the results.
ringerequiv	This test calculates the Ringer Equivalency Number (REN) for the telephone attached to the line. The test supports both the regular and electronic phone REN measurement techniques.
ringingselftest	This procedure verifies that the line circuit can generate high level differential signals such as those used during line testing or application of internally generated ringing to the loop. It generates a sinusoidal waveform with the requested amplitude and drives this signal into a test load of known resistance.

Table 233: TAC-ITM-RING Internal Line Tests (Continued)

Test ID	Description
ringingmonitor	This test is useful in checking the external ringing voltage given the loop cannot be disconnected while applying ringing and the ringing signal voltage cannot be reduced. This test is expected to be called on a line that has a terminating call (thus the need for applying ringing). This test uses about 3 cycles of the ringing waveform to carry out the test and then places the line to ringing state. Thus, a test is complete and we have placed ringing on the line as well to terminate the call. Please note that no ring trip would be detected during the first three cycles of the ringing signal.
tonegeneration	This test generates up to four sinusoidal tones simultaneously.
transhybridloss	This test measures trans-hybrid loss by generating a tone and measuring the reflected signal.
transmissionselftest	This procedure verifies that the line card can pass signals in the digital to analog and analog to digital directions. It measures trans-hybrid loss with open-circuit and a load impedance applied to the line. These trans-hybrid loss results are checked against expected values to generate a pass/fail result.

Metallic loop tests

This section outlines supported metallic loop tests, and provide some suggested boundary conditions as they are relevant to loop qualification:

- [3 elements capacitance test on page 1542](#)
- [3 elements insulation resistance test on page 1543](#)
- [DC feed self-test on page 1544](#)
- [DC loop resistance test on page 1545](#)
- [Distance to open test on page 1546](#)
- [DTMF and pulse digit measurement test on page 1546](#)
- [Foreign AC currents test on page 1548](#)
- [Foreign DC voltage test on page 1548](#)
- [Foreign AC voltage test on page 1549](#)
- [Howler test on page 1550](#)
- [Metering self test on page 1550](#)
- [Noise test on page 1551](#)
- [On-Off hook transition test on page 1551](#)
- [Loop and battery condition test on page 1552](#)
- [Receiver off-hook test on page 1553](#)
- [Ringer equivalency number test on page 1554](#)
- [Ringing self test on page 1554](#)

- [Ringing monitor test on page 1555](#)
- [Tone generation test on page 1555](#)
- [Trans-hybrid loss test on page 1556](#)
- [Transmission self test on page 1556](#)



Note: All the tests have the test time information as Time Started and Time Ended. The number listed in the Time Started and Time Ended are in hundredth of a second resolution. A typical test takes about 1.5 to 2 seconds.

3 elements capacitance test

The 3 elements capacitance test measures tip-to-ground (T-G), ring-to-ground (R-G), tip-to-ring (T-R) capacitance, and impedance.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/3/42 lookout 3elementcapacitance
Successful - In TestMode
Time Started: 334096
Time Ended: 334633
Three-Element capacitance Results
      (T-G) CAPACITANCE=      217.67 NFARADS
      (R-G) CAPACITANCE=      217.51 NFARADS
      (T-R) CAPACITANCE=      397.66 NFARADS
      (T-G) 55Hz AC IMPEDANCE= 13.01 KOHMS
      (R-G) 55Hz AC IMPEDANCE= 13.09 KOHMS
      (T-R) 55Hz AC IMPEDANCE= 7.23 KOHMS
-----
Successful - Returned to operational state
zSH>
```

3 elements capacitance test result description:

- (T-G) CAPACITANCE, (R-G) CAPACITANCE, (T-R) CAPACITANCE:
 - Reports the tip-to-ground, ring-to-ground, and tip-to-ring capacitances in nanofarads respectively. "NOT MEASURED" means the capacitance cannot be measured.
 - (T-R) CAPACITANCE value can be used to indicate whether there is a phone attached. In most the case, a capacitance less than 60 nF indicates there is no load. A value greater than 60 nF indicates there is a load attached, possibly a phone set.

NOTE: a modern phone with electronic ringer may have less than 60 nF between its Tip and Ring.

A "NOT-MEASURED" value in (T-R) CAPACITANCE may indicate the phone is off-hook. In this case, run the 3 element resistance test to verify the resistance value between Tip and Ring.

- (T-G) 55Hz AC IMPEDANCE, (R-G) 55Hz AC IMPEDANCE, 55Hz (T-R) AC IMPEDANCE:

Reports the tip-to-ground impedance at 55Hz in kOhms. "NOT MEASURED" means the impedance cannot be measured. If the result is less than 1200 kOhms, the actual measured value is displayed as the floating-point number. Otherwise, ">1200 KOHMS (OPEN)" is displayed.

3 elements insulation resistance test

The 3 elements insulation resistance test measures the resistance of an open loop. These measurements include resistance tip-to-ground (T-G), ring-to-ground (R-G), tip-to-ring (T-R); foreign DC voltage (T-G) and (R-G); and foreign DC current in the tip and ring leads.

Note that this test measures conditions on an open loop (T-R resistance is high). If the loop is closed, or the T-R resistance is under 2 kOhms, the DC loop resistance test should be used for a more accurate measurement.

Note that, the foreign DC voltage results from this test are not as accurate as those returned by the individual foreign DC voltage Test, but the overall testing time may be reduced by using results of this test instead of additionally running the foreign DC voltage test.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/17/4 lookout 3elementresistance
Successful - In TestMode
Time Started: 1129209
Time Ended: 1129456
Three-Element Resistance Results
      (T-G) DC RESISTANCE= > 1200          KOHMS (OPEN DC )
      (R-G) DC RESISTANCE= > 1200          KOHMS (OPEN DC )
      (T-R) DC RESISTANCE= 946.68          KOHMS
      (T-G) FOREIGN DC VOLTAGE= NONE       VOLTS
      (R-G) FOREIGN DC VOLTAGE= NONE       VOLTS
      TIP FOREIGN DC CURRENT= 0.00         MILLIAMPS
      RING FOREIGN DC CURRENT= 0.00         MILLIAMPS
-----
Successful - Returned to operational state
zSH>
```

3 elements resistance test result description:

- (T-G) DC RESISTANCE, (R-G) DC RESISTANCE, (T-R) DC RESISTANCE:
 - If the resistance is less than 150 ohms, it is considered to be very small, and interpreted as a short circuit or fault. Use DC loop resistance test for T-R resistance under 2 kOhms.
 - If the result could not be calculated because of some fault, NOT MEASURED is printed.

- If the resistance is larger than 1200 kOhms, it is considered to be too high to be measured accurately, and interpreted as open-circuit.
- Otherwise, the resistance result is considered as normal, and interpreted as a floating-point number.
- (T-G) FOREIGN DC VOLTAGE, (R-G) FOREIGN DC VOLTAGE:
 - If the result is less than 5 V, it is considered to be a normal value, and the foreign DC voltage is printed as a floating-point number.
 - If the result is between 5 V to 12 V, it is considered to be marginally normal, and printed as a floating-point number.
 - If the result is greater than 12 V, it is considered as not a normal value, and should be investigated.
 - If the result is printed as NONE, it means the result is normal for loop start, data loops, and CPE.
- TIP FOREIGN DC CURRENT, RING FOREIGN DC CURRENT:
 - If the result is less than 1 milliamps (mA), it is considered to be a normal value, and the tip foreign DC current is printed as a floating-point number.
 - If the result is between 1 mA to 3 mA, it is considered to be marginally normal, and printed as a floating-point number.
 - If the result is greater than 3 mA, it is considered as not a normal value, and should be investigated. If the result is greater than 8 mA, it is printed as “> 80 MILLIAMPS”, if the result is between 3 mA to 8 mA, the result is printed as floating-point number.
 - If the result is printed as NONE, it means the result is normal for loop start, data lines loops, and CPE.

DC feed self-test

This self test puts a 0.89 kOhms test load on the line, and measures the return in order to determine if appropriate levels are available on the line.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/4/1 lookout dcfeedselftest
Successful - In TestMode
Time Started: 9023093
Time Ended: 9023383
```

DC Feed Self-Test Results

TEST PASSED=	Yes	
MEASURED TEST LOAD=	892.34	OHMS
MEASURED HIGH BAT VOLTAGE=	-50.42	VOLTS
(T-R) MEASURED VOLTAGE=	40.87	VOLTS
CURRENT IN TEST LOAD (BAT=High, POL=Normal)=	16.21	MILLIAMPS
CURRENT IN TEST LOAD (BAT=High, POL=Reverse)=	16.30	MILLIAMPS
CURRENT IN TEST LOAD (BAT=Low, POL=Normal)=	15.69	MILLIAMPS

 Successful - Returned to operational state

DC feed self test result description:

- TEST PASSED indicates whether the test passed. It based solely on the measured test load, high battery potential and tip-ring voltage.
- MEASURED TEST LOAD reports the measured test load resistance. If this result is not within 10% of the nominal load resistance (0.89 kOhms), then TEST PASSED is set to NO.
- MEASURED HIGH BAT VOLTAGE reports the measured high battery voltage. It is nominal at -48 VDC. Acceptable ranges for this value are -42 to -56 VDC.
- (T-R) MEASURED VOLTAGE reports the measured tip-ring voltage while high current is driven. If the magnitude of the tip-ring voltage plus 2.66 V SLIC head-room plus $(90.4 / \text{MEASURED TEST LOAD}) \times (\text{T-R})$ MEASURED VOLTAGE is not within the MEASURED HIGH BAT VOLTAGE $\pm$ (1.1 V plus 10% of the magnitude of the high battery voltage), then TEST PASSED is set to NO.

DC loop resistance test

The DC loop resistance test measures the resistance on a line, longitudinal imbalances, and other characteristics. This test is useful for low loop resistance, generally less than 4 kOhms. For higher resistance loop the 3 elements resistance test is more accurate.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/4/1 lookout dcloopresistance
Successful - In TestMode
Time Started: 9025472
Time Ended: 9025648
```

DC loop resistance Test Results

```
LOOP RESISTANCE=                3.95 KOHMS
COMMON MODE CURRENT Phase 1=    0.00 MILLIAMPS
COMMON MODE CURRENT Phase 2=    0.00 MILLIAMPS
Voltage Saturation=              Yes
COMMON MODE CURRENT Degradation=No
```

 Successful - Returned to operational state

DC loop resistance test result description:

- LOOP RESISTANCE reports the measured loop resistance in kOhms.
- COMMON MODE CURRENT Phase 1 reports the common mode current measured during the test first phase.
- COMMON MODE CURRENT Phase 2 reports the common mode current measured during the test second phase.

- Voltage Saturation
 - = Yes indicates that the tip-ring voltage approached the battery voltage while attempting to drive the requested test current through the loop. The users should run the 3 element resistance test to get a more accurate measurement.
 - = No is a normal measurement.
- COMMON MODE CURRENT Degradation.
 - = Yes indicates that the test results may be inaccurate due to excessive common-mode current. The users should run the 3 element resistance test to get a more accurate measurement.
 - = No is a normal measurement.

Distance to open test

This test estimates the distance to an open-circuit by analyzing the results of a 3 elements resistance test and a 3 elements capacitance test.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/3/42 lookout distancetoopen
Successful - In TestMode
Time Started: 626510
Time Ended: 627395
Distance to open Results
          Distance to open=          4379.50          Meters
          Capacitance (measured)= 218.97          NFARADS
-----
Successful - Returned to operational state
zSH>
```

Distance to open test result description:

- Distance to open reports the estimated distance to an open-circuit in meters.
- Capacitance (measured) reports the measured capacitance value in nanofarads.
- If the test fails, one or both of the following errors will be displayed:
 - Test failed because the 3eleResistance failed – the Three-Element Insulation Resistance Test results show excessive current leakage.
 - Test failed because the 3eleCapcitence failed – the Three-Element Capacitance Test could not accurately measure the tip-to-ground and ring-to-ground capacitance.

DTMF and pulse digit measurement test

This test detects and measures a DTMF digit, pulse digit, or hook-switch flash. Only one digit or flash is reported for each invocation of this test. By

default, a single tone is output on the line during this test. The test runs for approximately 4 seconds.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/4/1 lookout
dtmfandpulsedigitmeasurement
Successful - In TestMode
Time Started: 9032539
Time Ended: 9032966
```

```
DTMF/pulse Results
DTMF/pulse test timed out
```

```
-----
Successful - Returned to operational state
```

DTMF and pulse digit measurement test result description:

- If no DTMF digits detected, the test result prints “DTMF/pulse test timed out”.
- If a DTMF digit was detected, the test result prints the real measurement as floating-point number.
- If a DTMF digit was detected and it has time to do a fourier transform the test result prints:
 - DTMF DIGIT <digit, e.g. 4>
 - DTMF SAMPLE SIZE= <number of FFT samples taken>
 - DTMF FIRST TONE= <frequency, e.g. 800> Hz
 - DTMF FIRST TONE LEVEL= <amplitude, e.g. 12> dBm
 - DTMF SECOND TONE= <frequency, e.g. 800> Hz
 - DTMF SECOND TONE LEVEL= <amplitude, e.g. 12> dBm
- If a Tone was detected but no DTMF digit detected and it has time to do a fourier transform it prints:
 - DTMF DIGIT NO DIGIT DECODED
 - DTMF SAMPLE SIZE= <number of FFT samples taken>
 - DTMF FIRST TONE= <frequency, e.g. 800> Hz
 - DTMF FIRST TONE LEVEL= <amplitude, e.g. 12> dBm
 - DTMF SECOND TONE= <frequency, e.g. 800> Hz
 - DTMF SECOND TONE LEVEL= <amplitude, e.g. 12> dBm
- If a pulse digit is detected it prints:
 - PULSE DIGIT= <digit, e.g. 7>
 - PULSE MINBRK= <% number> % OF AVG PERIOD
 - PULSE MAXBRK= <% number> % OF AVG PERIOD

- PULSE AVGBRK= <% number> % OF AVG PERIOD
- PULSE RATE= <number> PER SEC

(The MINBRK, MAXBRK, AVGBRK are a percentage of the average pulse period indicated by pulse rate.)

- If a hook flash is detected it prints “PULSE INTERVAL= <number e.g. 1120> mSEC”.
- If a disconnect is detected it prints “DISCONNECT DETECTED”.

Foreign AC currents test

This test measures foreign AC currents.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/4/1 lookout foreignaccurents
Successful - In TestMode
Time Started: 9035116
Time Ended: 9035207
```

```
Foreign AC current Results
TIP FOREIGN AC CURRENT= NONE          MILLIAMPS
RING FOREIGN AC CURRENT= NONE          MILLIAMPS
-----
```

Successful - Returned to operational state

The foreign AC currents test result description:

- TIP FOREIGN CURRENT reports the measured tip lead current in milliamps.
- RING FOREIGN CURRENT reports the measured ring lead current in milliamps.

Foreign DC voltage test

The foreign DC voltage test is examining the loop for the existence of DC voltage leaking onto a line from an external source.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/4/1 lookout foreigndcvoltage
Successful - In TestMode
Time Started: 9036757
Time Ended: 9036966
```

```
Foreign DC Voltage Test Results
Test Passed=Yes
(T-G) FOREIGN DC VOLTAGE= 0.02 VOLTS (OK)
(R-G) FOREIGN DC VOLTAGE= 0.00 VOLTS (OK)
(T-R) FOREIGN DC VOLTAGE= 0.02 VOLTS (OK)
-----
```

Successful - Returned to operational state

The foreign DC voltage test result description:

(T-G) FOREIGN DC VOLTAGE, (R-G) FOREIGN DC VOLTAGE, (T-R) FOREIGN DC VOLTAGE:

- <3 volts is normal. (OK) is printed after the measured data.
- >6 volts indicates a fault, it need to retest and follow-up. (FAULT) is printed after the measured data.
- = or > 100 volts indicates the presence of hazardous levels, and should be considered a dangerous fault. (HAZARDOUS) is printed after the measured data.
- For lines using ADSL2+, the voltage level for tip to ground should be less than 3 volts to ensure a stable DSL connection.

Foreign AC voltage test

The foreign AC voltage test is examining the loop for the existence of AC voltage leaking onto a line from an external source.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/4/1 lookout foreigndcvoltage
Successful - In TestMode
Time Started: 9038284
Time Ended: 9038424

Foreign AC Voltage Test Results
(T-G) FOREIGN AC VOLTAGE= 0.00 VRMS (NONE)
(R-G) FOREIGN AC VOLTAGE= 0.00 VRMS (NONE)
(T-R) FOREIGN AC VOLTAGE= 0.00 VRMS (NONE)
-----
Successful - Returned to operational state
```

Foreign AC Voltage test result description:

- (T-G) FOREIGN AC VOLTAGE, (R-G) FOREIGN AC VOLTAGE:
 - < 3 AC volts rms (Vrms), (NONE) will be printed out after a real measurement.
 - = or > 3 to = or < 10 AC Vrms is a normal and good measurement. It is normal for loop start, data lines loops, and CPE. (OK) will be printed out after a real measurement.
 - >10 to = or < 50 AC Vrms is a fault. It should be retested and then investigated. (FAULT) will be printed out after a real measurement.
 - >50 AC Vrms indicates the presence of hazardous levels, and should be considered a dangerous fault. (HAZARDOUS) will be printed out after a real measurement.
 - For lines using ADSL2+, the voltage levels for tip-to-ground and ring-to-ground should be less than 10 volts to ensure a stable DSL connection.

- (T-R) Foreign AC VOLTAGE:
 - < 3 Vrms is a normal and good measurement. It is normal for loop start, data lines loops, and CPE. (NONE) will be printed out after a real measurement.
 - =or >3 to = or < 5 Vrms is a fault. It should be retested and then investigated. (FAULT) will be printed out after a real measurement.
 - >50 Vrms indicates the presence of hazardous levels, and should be considered a dangerous fault. (HAZARDOUS) will be printed out after a real measurement.
 - For lines using ADSL2+, the voltage level for tip to ring should be less than 3 volts to ensure a stable DSL connection.

Howler test

This procedure generates a Howler (Receiver Off-Hook) tone until the phone goes on-hook or a timeout condition is detected.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/4/1 lookout howlertest
Successful - In TestMode
Time Started: 9039942
Time Ended: 9040152
Howler Test results
Running US Howler Test
-----
Successful - Returned to operational state
```

The howler test result description:

Depending on the system profile, the howler test prints “Running US Howler Test”, “Running Australian Howler Test”, or “Running UK Howler Test”. If the system profile cannot be read, the test prints “Failed to access the system profile”, and stop the test.

Metering self test

This procedure verifies that the line card can generate a metering pulse. It drives a metering signal into both a resistive load and an open-circuit using the current Metering Profile applied to the line.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/3/42 lookout meteringselftest
Successful - In TestMode
Time Started: 396242
Time Ended: 396298
Metering Self-Test Results
TEST PASSED= Yes
Peak metering Voltage Resistive Load= 1.22 VOLTS
Peak metering Voltage Open circuit= 1.40 VOLTS
```

```
-----
Successful - Returned to operational state
zSH>
```

The metering self test result description:

- Peak metering Voltage Resistive Load reports the peak voltage of the metering signal with the circuit connected to a resistive load.
- Peak metering Voltage Open Circuit reports the peak voltage of the metering signal with the circuit open.

Noise test

The noise test measures the amount of noise in dBm on the line, relative to TLP 0. This provides measurements in dBm0 units.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/4/1 lookout noisemeasurement
Successful - In TestMode
Time Started: 9047559
Time Ended: 9047703
```

```
Noise Test results
NOISE=          -67.23          dBm0
-----
```

```
Successful - Returned to operational state
```

Noise test result description:

- Noise below -45 dBmO is an average loop (LSB switching noise approaches -45 dBmO).
- Noise between -44 and -10 dBmO is too noisy and should be retested and investigated.

On-Off hook transition test

This on-hook to off-hook test allows the MLT to determine if a loop can successfully complete a simulated hook state transition.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/4/1 lookout onandoffhookmeasurement
Successful - In TestMode
Time Started: 9049153
Time Ended: 9049231
```

```
ON-OFF Hook Self-Test Results
PASSED
-----
```

```
Successful - Returned to operational state
```

The on-off hook transition test result description:

- PASSED indicates that the test was passed.
- ABORTED indicates that the line was off-hook when the test was started.
- HW_FAULT indicates that the test failed because the line circuit did not properly detect on-hook and off-hook state changes.
- UNKNOWN indicates some unexpected error occurred.

Loop and battery condition test

The loop and battery condition test measures the instantaneous loop resistance, loop currents, and loop and battery voltages. No filtering is done during the measurement, so the results may fluctuate from one reading to the next in the presence of AC induction on the line.

The following example tests the POTS line on shelf 1, slot 4, port 1 with a forced readloopandbatteryconditions test using lookout mode, and provides the outputs:

```
zSH> mtac-linetest 1/4/1 lookout readloopandbatteryconditions force
Successful - In TestMode
Time Started: 9053736
Time Ended: 9053737

Read Loop and Battery Condition TestResults
Read Loop and Battery Condition Test
      LOOP resistance=                not measured
      Common-mode (longitudinal) current=    not measured
      (T-R) (metallic) current=            not measured
      (T-R) voltage=                    not measured
      Lowest battery voltage (measured)=      -50.30          VOLTS
      Highest battery voltage (measured)=     -50.41          VOLTS
      Positive battery voltage=              1.40            VOLTS
      Metering Voltage (measured)=           0.00            VOLTS
      NOTE: the metering voltage is only valid if a metering pulse is currently
            being generated.
-----
Successful - Returned to operational state
```

Loop and battery condition test result description:

- Loop resistance reports the loop resistance in kOhms. If the loop resistance cannot be measured in the present line state, “not measured” is reported.
- Common-mode (longitudinal) current reports the longitudinal (common-mode) current in milliamps. If the longitudinal current cannot be measured in the present line state, “not measured” is reported.
- (T-R) (metallic) current reports the metallic (tip-to-ring) current in milliamps. If the metallic current cannot be measured in the present line state, “not measured” is reported.

- (T-R) voltage reports the tip-to-ring voltage in volts. If the tip-to-ring voltage cannot be measured in the present line state, "not measured" is reported.
- Lowest battery voltage (measured) reports the voltage of battery with the lowest absolute value in volts.
- Highest battery voltage (measured) reports the voltage of the battery with the highest absolute value in volts.
- Positive battery voltage reports the positive battery voltage in volts.
- Metering Voltage (measured) reports the peak metering signal voltage observed across tip and ring since the start of the metering pulse.

Receiver off-hook test

The receiver off-hook test allows the MLT to determine if a loop has one or more telephones that are off-hook at the far end of the circuit.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/4/1 lookout receiveroffhook
Successful - In TestMode
Time Started: 9057457
Time Ended: 9057635
```

Receiver Off-Hook Test Results

```
OFF-HOOK= No
RLOOP out of range= Yes
```

Successful - Returned to operational state

Receiver off-hook test results are described in the following table:

Table 234: Receiver off-hook test result description

OFF-HOOK	RLOOP out of range	Test Result Description
NO	NO	The DC Loop Resistance Test returned real resistance values, but they are not characteristic of an off-hook receiver.
YES	NO	This test measured loop resistances that suggest an off-hook receiver.
NO	YES	The DC Loop Resistance Test failed to measure loop resistance or returned an unreasonable result. This is most likely due to the receiver being on-hook.
YES	YES	This test never returns this result.

Ringer equivalency number test

This test calculates the Ringer Equivalency Number (REN) for the telephone attached to the line. The test supports both the regular and electronic phone REN measurement techniques.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/3/42 lookout ringerequiv
Successful - In TestMode
Time Started: 415643
Time Ended: 415740
Ringer Equivalence Number Test Results
      REN=                                0.49          RINGEQIV
      Measured Zload=                     14.26          KOHMS
      COMMON MODE CURRENT Degradation=    no
-----
Successful - Returned to operational state
zSH>
```

Ringer equivalency number test result description:

- REN reports the measured Ringer Equivalency Number (REN).
- Measured Zload reports the measured ringer impedance in kOhms and only applies to the regular phone REN test. It is set to zero if the application ran an electronic phone REN test.
- COMMON MODE CURRENT Degradation is YES if the test results may be inaccurate due to excessive common-mode current. This flag only applies to the regular phone REN test and is set to zero if the application ran an electronic phone REN test.

Ringing self test

The ringing self test is a simulation of ringing on current able to be passed on the line. As a note, no actual ringing will be audible due to low voltage used.

The following example provide the sample command and output:

```
zSH> mtac-linetest 1/4/1 lookout ringingselftest
Successful - In TestMode
Time Started: 9061926
Time Ended: 9062086
Ringing Self-Test Results
      TEST PASSED=                        Yes
      RLOOP=                             2.53          KOHMS
      ON HOOK TO OFF HOOK TRANSITION DETECTED= Yes
-----
Successful - Returned to operational state
```

This test is informational, and is used to determine if loop conditions on the line will allow ringing current to reach the far end of the circuit. This test should pass on valid loops.

Ringling monitor test

The ringing monitor test checks the external ringing voltage given the loop cannot be disconnected while supplying ringing and the ringing signal voltage cannot be reduced.

The following example provides the sample command and output:

```
zSH> mtac-linetest 1/4/1 lookout ringingmonitor
Successful - In TestMode
Time Started: 9078335
Time Ended: 9078393
```

Ringling Monitor Results

```
Test Aborted due to off hook=No
Ring Voltage= 0.00 VRMS (NONE)
```

```
-----
Successful - Returned to operational state
```

Ring monitor test result description:

- Test abort due to off hook indicates whether the test was aborted due to off-hook detection at the beginning of the test.
- Ring voltage reports the measured external ringing voltage in RMS volts.

Tone generation test

This test generates up to four sinusoidal tones simultaneously. It also provides the optional parameters to set the maximum duration and frequency of the tone. By default, the duration is 2 seconds, and frequency is 1000 Hz.

The following example provides the sample commands, and the outputs for the succeeded tests.

A basic tone generation test:

```
zSH> mtac-linetest 1/4/1 lookout tonegeneration
Successful - In TestMode
Time Started: 9079951
Time Ended: 9080179
```

```
-----
Successful - Returned to operational state
```

A tone generation test with the maximum duration of 60 seconds.

```
zSH> mtac-linetest 1/4/1 lookout tonegeneration 60
Successful - In TestMode
Time Started: 3127104
Time Ended: 3133173
```

```
-----
Successful - Returned to operational state
```

A tone generation test with the maximum duration of 60 seconds and tone frequency of 2000 Hz.

```

zSH> mtac-linetest 1/4/1 lookout tonegeneration 180 2000
Successful - In TestMode
Time Started: 3135884
Time Ended: 3154046

```

```

-----
Successful - Returned to operational state

```

The tone generation tests in the above examples are succeed although in the output didn't show the data.

Trans-hybrid loss test

This loop test characterizes the amount of echo from a far end trans-hybrid unit. This is only found in a telephone device, and is not a valid test on a dry pair for DSL.

The following example provides the sample command and output:

```

zSH> mtac-linetest 1/4/1 lookout transhybridloss
Successful - In TestMode
Time Started: 9081809
Time Ended: 9081938

```

Transhybrid Loss Results

ECHO=	-84.13	dBm0
LOSS=	74.13	dBm0

```

-----
Successful - Returned to operational state

```

Trans-hybrid loss test result description:

- ECHO returns the measured signal echo power in dBm0.
- LOSS returns the calculated trans-hybrid loss in dB.

Transmission self test

The transmission self test attempts to determine if the line's trans-hybrid loss using a test load is greater than an allowed minimum loss. The test load value should be greater than the lower limit value. A trans-hybrid device is only found in a telephone device, and is not a valid test on a dry pair for DSL.

The following example provides the sample command and sample output:

```

zSH> mtac-linetest 1/3/42 lookout transmissionselftest
Successful - In TestMode
Time Started: 429667
Time Ended: 429743

```

Transmission Self Test Results

TEST PASSED=	No	
TEST ABORT, OFF_HOOK=	No	
TRANS-HYBRID LOSS OPEN=	53.30	dB
TRANS_HYBRID LOSS RLOAD=	8.98	dB
EXPECTED TRANS-HYBRID LOSS LOWER LIMIT=	20.83	dB
EXPECTED TRANS-HYBRID LOSS UPPER LIMIT=	VERY_HIGH_THL	

 Successful - Returned to operational state

This test is informational, and is used to determine trans-hybrid loss on a loop. This test should pass on valid loops. The loss on open circuits should read nominally 0. The loss on the test load should read higher than the loss lower limit. If the measured level is lower than stated lower limit, then it may indicate a problem with the line.

Troubleshooting with metallic loop tests

To diagnose the problem in the metallic loop, may takes several different TAC tests. The following examples provide the sample troubleshooting cases.

Phone is off-hook

To troubleshoot whether the phone is off-hook, use the 3 element capacitance test and 3 element resistance test. The (T-R) CAPACITANCE value can be used to indicate whether there is a phone attached. In most cases, a capacitance less than 60 nanofarads indicates the Tip to Ring is open, there is no load (e.g. no phone attached); A value greater than 60 nanofarads indicates there is a load attached, possibly a phone set; A value “NOT MEASURED” indicates the Tip to Ring is shorted, and possibly the phone is off-hook.



Note: The following examples in this section are not using the modern phone.

A modern phone with electronic ringer may have less than 60 nanofarads between its Tip and Ring.

Here is an example of phone is off-hook (with 9600 ft. cable):

- 1 At first, look the (T-R) CAPACITANCE value in the 3 element capacitance test output. A “NOT-MEASURED” value in T-R CAPACITANCE indicate the phone is possibly off-hook.

```
zSH> mtac-linetest 1/7/27 lookout 3elementcapacitance force
```

Three-Element capacitance Results

(T-G)	CAPACITANCE=	155.09	NFARADS
(R-G)	CAPACITANCE=	156.71	NFARADS
(T-R)	CAPACITANCE=	NOT MEASURED	
(T-G)	55Hz AC IMPEDANCE=	16.10	KOHMS
(R-G)	55Hz AC IMPEDANCE=	15.97	KOHMS
(T-R)	55Hz AC IMPEDANCE=	NOT MEASURED	

- 2 Then run the 3 element resistance test to verify the resistance value between Tip and Ring. The “748.47 OHMS” value in (T-R) DC RESISTANCE indicates the Tip and Ring are closed or shorted. Based on this information, then we can diagnosed that the phone is off-hook.

```
zSH> mtac-linetest 1/7/27 lookout 3elementresistance force
```

Three-Element Resistance Results

(T-G) DC RESISTANCE=	> 1200	KOHMS (OPEN DC)
(R-G) DC RESISTANCE=	> 1200	KOHMS (OPEN DC)
(T-R) DC RESISTANCE=	748.47	OHMS
(T-G) FOREIGN DC VOLTAGE=	NONE	VOLTS
(R-G) FOREIGN DC VOLTAGE=	NONE	VOLTS
TIP FOREIGN DC CURRENT=	0.00	MILLIAMPS
RING FOREIGN DC CURRENT=	0.00	MILLIAMPS

Phone is on-hook

Here is an example of phone is on-hook (with 9600 ft. 24 AWG cable):

Run the 3 element capacitance test. Look the (T-R) CAPACITANCE value in the 3 element capacitance test output. In this example, the value “124.67 NFARADS” is greater than 60 nanofarads, it indicates the phone is on-hook.

```
zSH> mtac-linetest 1/7/27 lookout 3elementcapacitance force
```

Three-Element capacitance Results

(T-G) CAPACITANCE=	151.11	NFARADS
(R-G) CAPACITANCE=	151.75	NFARADS
(T-R) CAPACITANCE=	124.67	NFARADS
(T-G) 55Hz AC IMPEDANCE=	16.52	KOHMS
(R-G) 55Hz AC IMPEDANCE=	16.49	KOHMS
(T-R) 55Hz AC IMPEDANCE=	20.21	KOHMS

Phone is not attached

Here is an example of no phone is attached:

Run the 3 element capacitance test. Look the (T-R) CAPACITANCE value in the 3 element capacitance test output. In this example, the value “0.79 NFARADS” is less than 60 nanofarads, it indicates the Tip to Ring is open, there is no load (e.g. no phone attached).

```
zSH> mtac-linetest 1/7/27 lookout 3elementcapacitance force
```

Three-Element capacitance Results

(T-G) CAPACITANCE=	1.62	NFARADS
(R-G) CAPACITANCE=	1.59	NFARADS
(T-R) CAPACITANCE=	0.79	NFARADS
(T-G) 55Hz AC IMPEDANCE=	508.99	KOHMS
(R-G) 55Hz AC IMPEDANCE=	552.75	KOHMS
(T-R) 55Hz AC IMPEDANCE=	> 1200	KOHMS (OPEN)

Both Tip and Ring are grounded

Here is an example where the loop line is grounded on both Tip and Ring. In this case, both 3 element resistance and 3 element capacitance tests would fail, indicating the line is shorted and grounded. With the DC loop resistance test, it may be possible to use the loop resistance value to determine the distance of the line.

- 1 At first, run the 3 element capacitance test. “NOT-MEASURED” values indicate cannot measure these values in the loop line.

```
zSH> mtac-linetest 1/7/26 lookout 3elementcapacitance
force
```

Three-Element capacitance Results

(T-G) CAPACITANCE=	NOT MEASURED
(R-G) CAPACITANCE=	NOT MEASURED
(T-R) CAPACITANCE=	NOT MEASURED
(T-G) 55Hz AC IMPEDANCE=	NOT MEASURED
(R-G) 55Hz AC IMPEDANCE=	NOT MEASURED
(T-R) 55Hz AC IMPEDANCE=	NOT MEASURED

- 2 Then run the 3 element resistance test. Look the DC RESISTANCE value in the 3 element resistance test output. A “< 150” value is considered to be very small, and interpreted as a short circuit or fault.

```
zSH> mtac-linetest 1/7/26 lookout 3elementresistance force
```

Three-Element Resistance Results

(T-G) DC RESISTANCE= < 150	OHMS (FAULT) TEST HALTED
(R-G) DC RESISTANCE= < 150	OHMS (FAULT) TEST HALTED

- 3 And then run the DC loop resistance test with an 100 feet cable.

```
zSH> mtac-linetest 1/7/26 lookout dcloopresistance force
```

DC loop resistance Test Results

LOOP RESISTANCE=	0.07	KOHMS
COMMON MODE CURRENT Phase 1=	0.00	MILLIAMPS
COMMON MODE CURRENT Phase 2=	0.00	MILLIAMPS
Voltage Saturation=	No	
COMMON MODE CURRENT Degradation=	Yes	

- 4 Or run the DC loop resistance test with a 9600 feet 24 awg cable.

```
zSH> mtac-linetest 1/7/26 lookout dcloopresistance force
```

DC loop resistance Test Results

LOOP RESISTANCE=	0.56	KOHMS
COMMON MODE CURRENT Phase 1=	0.00	MILLIAMPS
COMMON MODE CURRENT Phase 2=	0.00	MILLIAMPS
Voltage Saturation=	No	
COMMON MODE CURRENT Degradation=	Yes	

Only Ring wire is grounded

The following example shows how to use metallic test command to diagnose the Ring to Ground (R-G) is shorted, yet, the Tip to Ground (T-G) is open.

- 1 At first, run the 3 element capacitance test. “NOT-MEASURED” values indicate cannot measure these values in the loop line.

```
zSH> mtac-linetest 1/7/26 lookout 3elementcapacitance
force
```

Three-Element capacitance Results		
(T-G)	CAPACITANCE=	NOT MEASURED
(R-G)	CAPACITANCE=	NOT MEASURED
(T-R)	CAPACITANCE=	NOT MEASURED
(T-G)	55Hz AC IMPEDANCE=	NOT MEASURED
(R-G)	55Hz AC IMPEDANCE=	NOT MEASURED
(T-R)	55Hz AC IMPEDANCE=	NOT MEASURED

- 2 Then run the 3 element resistance test. It only shows (R-G) DC RESISTANCE value, didn't show the (T-G) DC RESISTANCE value, and the value is "< 150". This indicates Ring to Ground is shorted, Tip to Ground is open.

```
zSH> mtac-linetest 1/7/26 lookout 3elementresistance force
```

Three-Element Resistance Results		
(R-G)	DC RESISTANCE= < 150	OHMS (FAULT) TEST HALTED

Auto-calibration

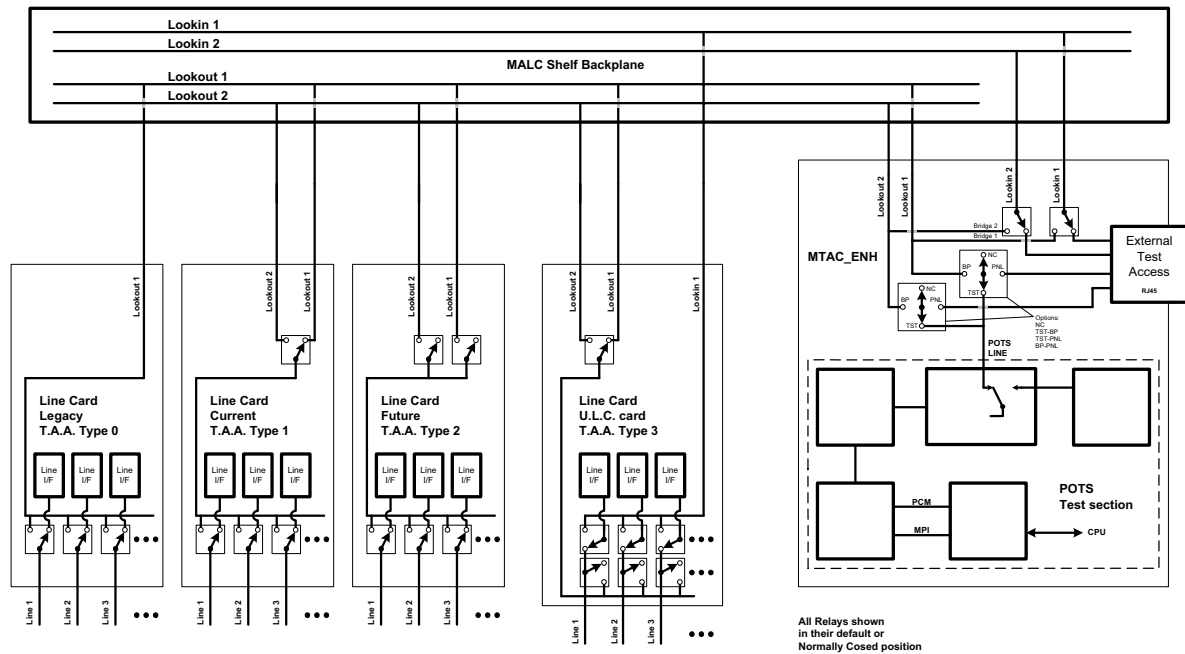
When the **mtac-linetest** command is issued, prior to running the line test, the line card performs an auto-calibration.

Lookout block diagram

Figure 183: Lookout block diagram

AJK 2007-05-23

MALC Shelf Test Attach Architecture (T.A.A.) Block Diagram



Configuring external alarms

The TAC cards have a 26 pin connector that provides sensing of alarm relay contacts for up to 12 external devices. When an alarm condition occurs on the external device, the MXK sends a trap. Each of the 12 pairs of pins can be assigned to a different alarm. Use the **num2str-profile** to assign a description to an alarm relay. The description is included in traps and log messages.

The **num2str-profile** uses an index in the form:

shelf/slot/282/alarm-contact

The following example adds a description to the first alarm contact of a TAC-RING card in *shelf* 12:

```
zSH> update num2str-profile 1/12/282/1
Please provide the following: [q]uit.
name: -> {Relay 1}: cabinet open
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

Configuring an external clock

The MXK supports the following external clock sources:

- A recovered clock from a T1/E1 line.
- Building Integrated Timing Supply (BITS) clock.

Connecting a T1/E1 recovered clock to the TAC card

The network T1/E1 clock on the TAC card appears to the system as a T1/E1 interface. To connect the clock source:

- 1 Connect the T1/E1 cable to the TAC card external clock input port which is an RJ-45 port labeled CLOCK. This is used to source the clock to the shelf using standard T1/E1 pin connections.
- 2 Configure the system to use the clock, as explained in [MXK Clocking on page 173](#).

Connecting a BITS clock to the TAC card

The external clock input port on the TAC card uses pins 6 and 8 for ground and pin 7 for the clock reference. To connect the BITS clock (also known as 2 Mhz clock) source:

- 1 Connect the BITS clock signal to the TAC card external clock input port which is an RJ-45 port labeled CLOCK, pin 7.
- 2 Connect the clock ground line to both pins 6 and 8 together (pin 6 is the 2 MHz input selector pin). This selects use of the 2 MHz BITS clock instead of the T1/E1 recovered clock.

3 Configure the **card-line-type** parameter in the TAC **card-profile** to **e1**.

```
zSH> card add 14 linetype e1 group 2
```

An autogenerated card-group-id [2] is assigned for this card type.

new card-profile 1/14/5072 added, sw-file-name "tacitmring.bin", 2 options:

```
card-group-id 2 card-line-type e1
```

Verify the card-line-type:

```
zSH> get card-profile 1/14/5072
```

```
card-profile 1/14/5072
```

Please provide the following: [q]uit.

```
sw-file-name: -----> {tacitmring.bin}:
```

```
admin-status: -----> {operational}:
```

```
upgrade-sw-file-name: ---> {}:
```

```
upgrade-vers: -----> {}:
```

```
admin-status-enable: ----> {enable}:
```

```
sw-upgrade-admin: -----> {reloadcurrrev}:
```

```
sw-enable: -----> {true}:
```

```
sw-upgrade-enable: -----> {false}:
```

```
card-group-id: -----> {0}:
```

```
hold-active: -----> {false}:
```

```
weight: -----> {nopreference}:
```

```
card-line-type: -----> {e1}:
```

```
card-atm-configuration: -> {notapplicable}:
```

```
card-line-voltage: -----> {not-used}:
```

```
maxvpi-maxvci: -----> {notapplicable}:
```

```
card-init-string: -----> {}:
```

```
wetting-current: -----> {disabled}:
```

```
pwe-timing-mode: -----> {none}:
```

```
.....
```

Save new record? [s]ave, [c]hange or [q]uit: **s** An autogenerated card-group-id [2] is assigned for this card type.

New record saved.

4 When the TAC card is running, update **ds1-profile** by specifying the **line-type** parameter to **other** and the **transmit-clock-source** to **looptiming**.

```
zSH> update ds1-profile 1-14-1-0/ds1
```

```
ds1-profile 1-14-1-0/ds1
```

Please provide the following: [q]uit.

```
line-type: -----> {e1crc}: other
```

```
line-code: -----> {hdb3}:
```

```
send-code: -----> {sendnocode}:
```

```
circuit-id: -----> {e1}:
```

```
loopback-config: -----> {nolooop}:
```

```
signal-mode: -----> {none}:
```

```
fdl: -----> {fdlnone}:
```

```
dsx-line-length: -----> {dsx0}:
```

```
line-status_change-trap-enable: ---> {enabled}:
```

```
channelization: -----> {disabled}:
```

```
ds1-mode: -----> {other}:
```

```

csu-line-length: -----> {csu00}:
clock-source-eligible: -----> {eligible}:
transmit-clock-source: -----> {throughtiming}: looptiming
cell-scramble: -----> {true}:
coset-polynomial: -----> {true}:
protocol-emulation: -----> {network}:
signal-type: -----> {loopstart}:
dsl-group-number: -----> {0}:
line-power: -----> {disabled}:
timeslot-assignment: -----> {0}:
transmit-clock-adaptive-quality: --> {stratum3}:
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.

```

- 5 Configure the system to use the clock, as explained in [MXK Clocking on page 173](#).

Connecting an external ring source

The TAC card provides support for an external ring source to provide ringing voltage for the system.



Caution: When connecting the external ring source, observe the following:

If the external ring generator is an internal -48V reference (non-isolated), connect the top pin to the ringing voltage using a minimum 22 AWG wire, and leave the bottom pin unconnected. See [Figure 184 on page 1565](#).

If the external ring generator requires an external -48V reference (isolated), connect the top pin to the ringing voltage using a minimum 22 AWG wire and the bottom pin to -48V on the ring source. See [Figure 185 on page 1566](#).

Figure 184: Connecting a non-isolated ring source

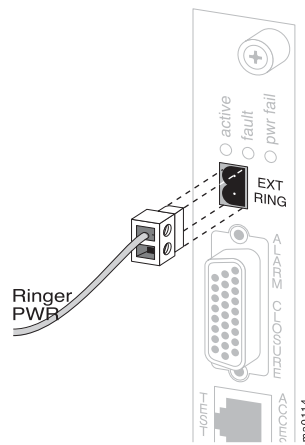
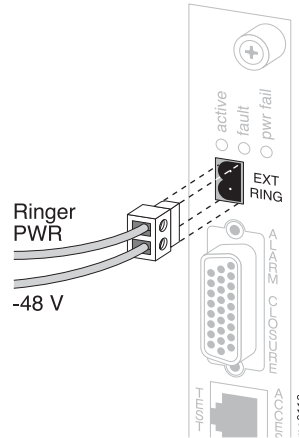


Figure 185: Connecting an isolated ring source



After connecting the ring source, update the **system** profile to specify an external ring source:

```
zSH> update system 0

system 0
Please provide the following: [q]uit.
syscontact: -----> {Zhone Global Services and Support 7001 Oakport Road Oakland Ca. (877) Zhone20 (946-6320) Fax (510) 777-7113 support@zhone.com}:
sysname: -----> {Zhone Mxk}:
syslocation: -----> {Oakland}:
enableauthtraps: -----> {disabled}:
setserialno: -----> {0}:
zmsexists: -----> {false}:
zmsconnectionstatus: -> {inactive}:
zmsipaddress: -----> {0.0.0.0}:
configsyncexists: -----> {false}:
configsyncoverflow: --> {false}:
configsyncpriority: --> {high}:
configsyncaction: -----> {noaction}:
configsyncfilename: --> {}:
configsyncstatus: -----> {syncinitializing}:
configsyncuser: -----> {}:
configsyncpasswd: -----> {}:
numshelves: -----> {1}:
shelvesarray: -----> {}:
numcards: -----> {3}:
ipaddress: -----> {0.0.0.0}:
alternateipaddress: --> {0.0.0.0}:
countryregion: -----> {us}:
primaryclocksource: --> {0/0/0/0/0}:
ringsource: -----> {internalringsourcelabel}: externalringsourcelabel
revertiveclocksource: -> {true}
voicebandwidthcheck: --> {false}
alarm-levels-enabled:--> {critical+major+minor+warning}
userauthmode:-----> {local}
```

```
radiusauthindex:-----> {0}
secure:-----> {disabled}
webinterface:-----> {enabled}
options:-----> {NONE(0) }
.....
Save changes? [s]ave, [c]hange or [q]uit: s
Record updated.
```

TAC cards pinouts

This section lists the pinouts for the following interfaces on the TAC cards:

- [External ring generator input port pinouts](#)
- [External alarm sense pinouts](#)
- [Examples of alarms with specific pinouts](#)
- [Metallic test access port pinouts](#)
- [External test set control port pinouts](#)
- [External clock input port pinouts](#)

External ring generator input port pinouts

The TAC cards provide an external ring generator input port for access to external ring generator.

Figure 186: TAC-ITM-RING card external ring generator input connector pinouts

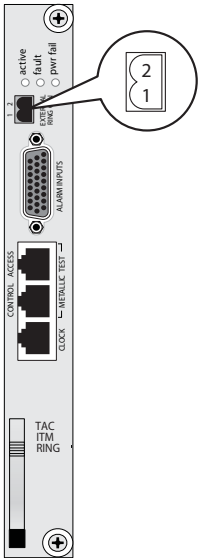


Table 235 lists the pinouts for the external ring generator.

Table 235: External ring generator pinouts

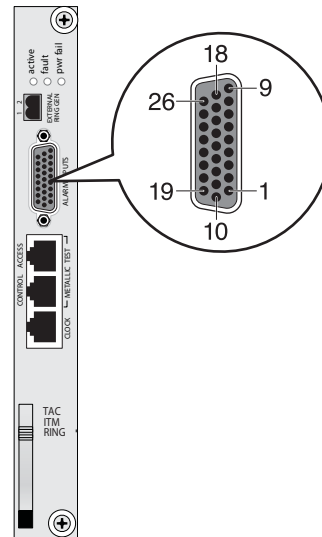
Pin numbering for TAC-RING and TAC-ITM-RING	Function
2	Ring Power Input
1	-48V Output

External alarm sense pinouts

The TAC cards provide a 26-pin connector for access to external alarms.

The TAC cards accept 48-volt inputs directly. All alarm inputs are metalically isolated using optocouplers. All TAC-ITM-RING cards take 48 volts directly. Check with Zhong GSS for use of alarm sense contacts on Revision L or earlier MTAC/RING cards.

Figure 187: TAC-ITM-RING card external alarm connector pinouts



[Table 236](#) lists the pinouts for the 26-pin connector for access to external alarms.

Table 236: TAC card external alarm connector pinouts

External alarm	Pin	Function
N/A	1	-48V supply for external contacts (fused)
1	2	Input (+)
	3	Input (-)
2	4	Input (+)
	5	Input (-)
3	6	Input (+)
	7	Input (-)
4	8	Input (+)
	9	Input (-)

Table 236: TAC card external alarm connector pinouts (Continued)

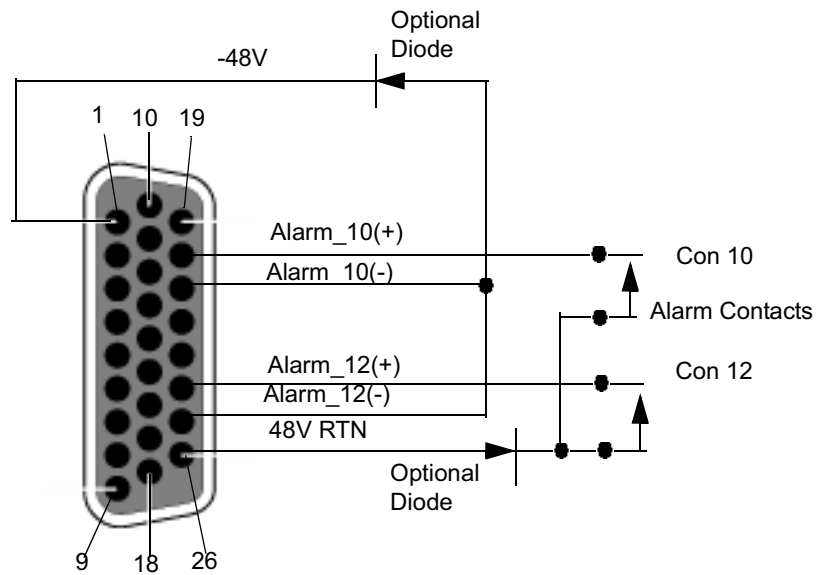
External alarm	Pin	Function
5	10	Input (+)
	11	Input (-)
6	12	Input (+)
	13	Input (-)
7	14	Input (+)
	15	Input (-)
8	16	Input (+)
	17	Input (-)
9	18	Input (+)
	19	Input (-)
10	20	Input (+)
	21	Input (-)
11	22	Input (+)
	23	Input (-)
12	24	Input (+)
	25	Input (-)
N/A	26	48V return (+)

Examples of alarms with specific pinouts

The following example shows alarms 10 and 12 for a single TAC-ITM-RING. See [Table 236](#) for other alarm pin numbers.



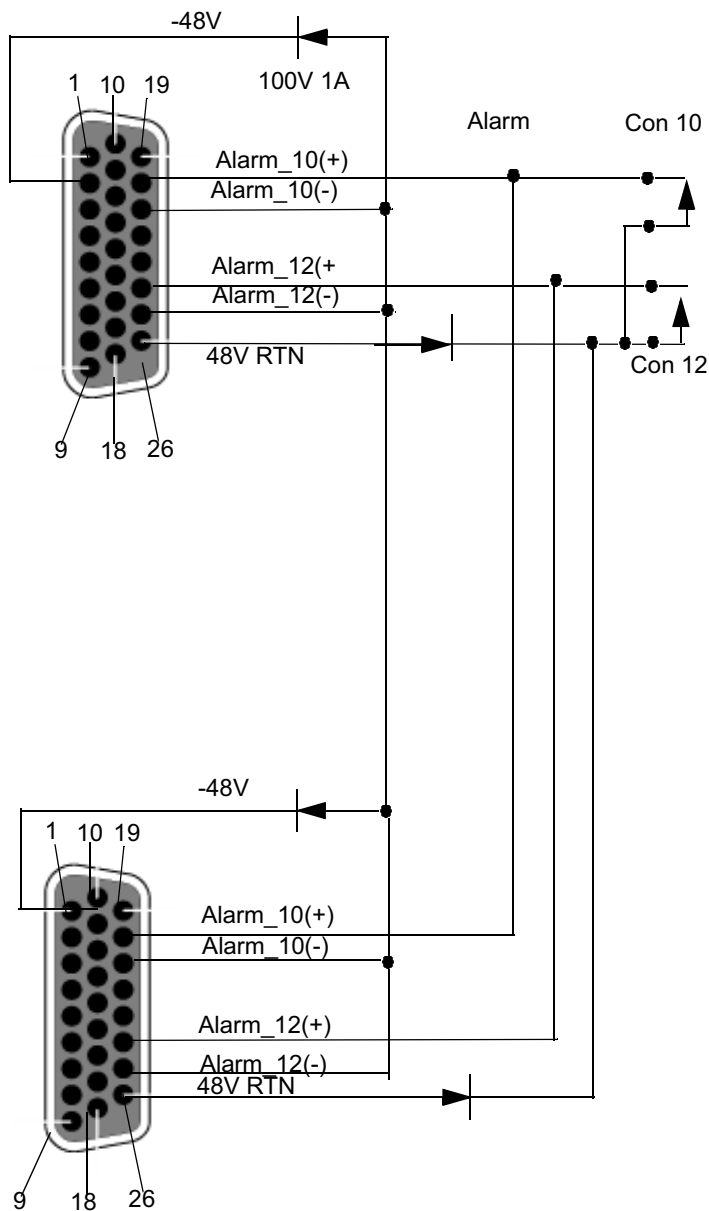
Note: Diodes is optional in single card case.

Figure 188: Single TAC-ITM-RING Sample Connections

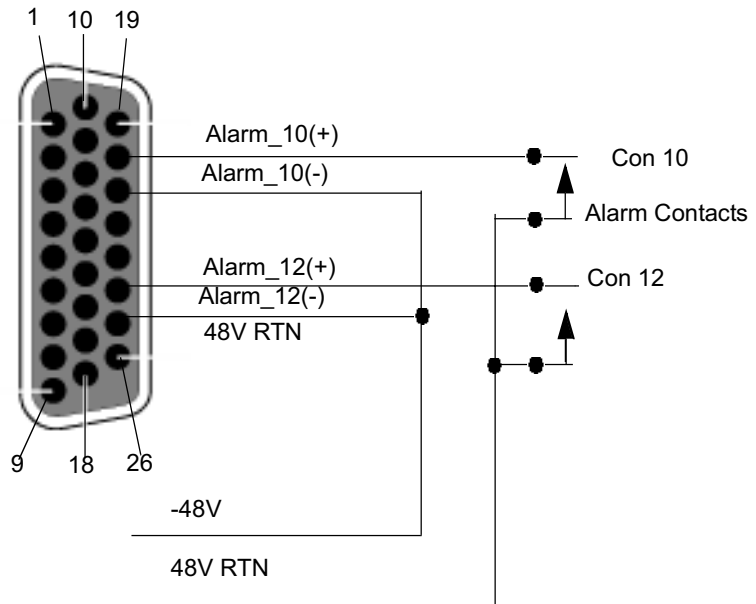
The following example shows alarms 10 and 12 for a redundant TAC cards, using board-supplied contact voltage. See [Table 236](#) for other alarm pin numbers.

Figure 189: Redundant TACs: Example Connections

In 4002 or Equivalent (4 Places)

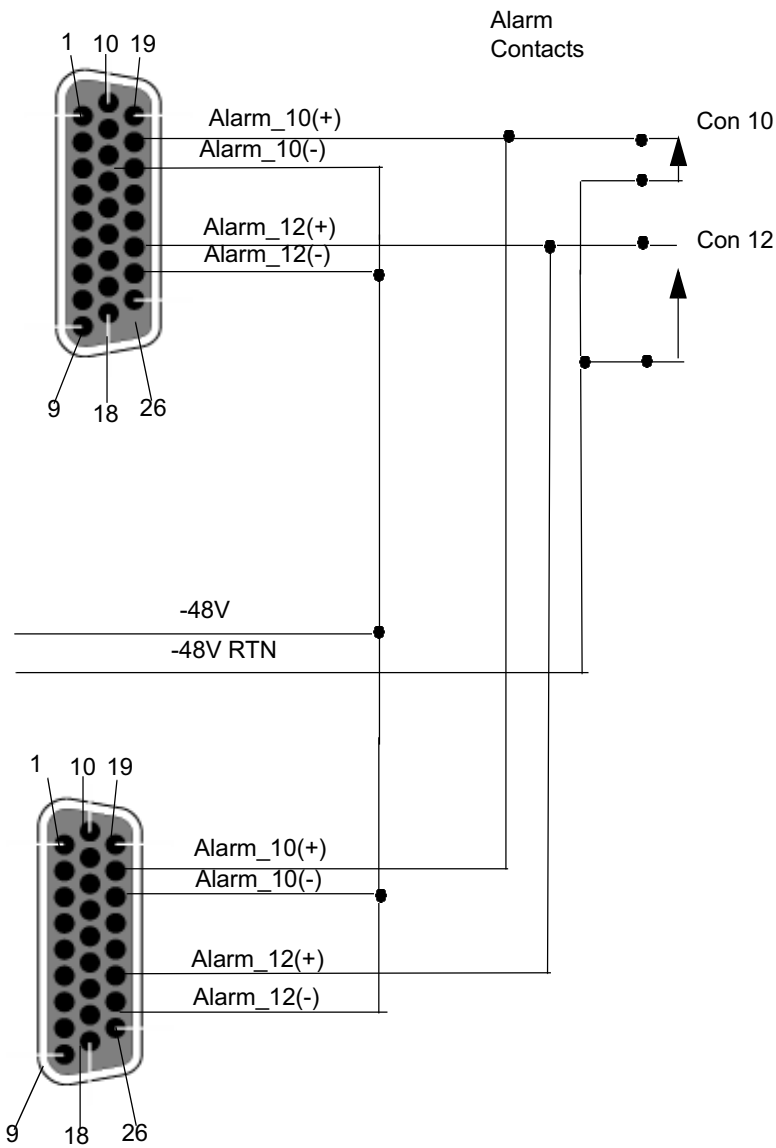


The following example shows alarms 10 and 12 for a single TAC-ITM-RING. See [Table 236](#) for other alarm pin numbers.

Figure 190: Single TAC-ITM-RING with Externally Supplied Contact Voltage

The following example shows alarms 10 and 12 for redundant TAC cards with an externally supplied contact voltage.

Figure 191: Redundant TACs: Any Combination of TAC-ITM-RING with Externally Supplied Contact Voltage



Metallic test access port pinouts

The TAC cards provide a metallic test access port for access to an external test set.

Figure 192: TAC-ITM-RING card metallic test access port pinouts

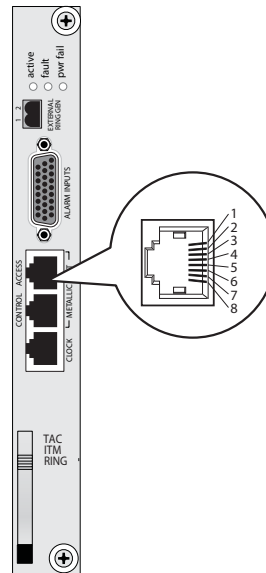


Table 237 lists the pinouts for the TAC card metallic test access port.

Table 237: TAC card metallic test access port

Pin	Function
1	Test in tip 1
2	Test in ring 1
3	Test out tip 1
4	Test out ring 1
5	Test in tip 2
6	Test in ring 2
7	Test out tip 2
8	Test out ring 2

External test set control port pinouts

The TAC cards provide an external test set control port to provide a control connection to the external test set.

[Table 238](#) lists the pinouts for the TAC card external test RS232 control port.

* Factory test signals do not connect on TAC-ITM-RING.

Table 238: TAC card external test control port pinouts

Pin	Function
1	*Reserved
2	*Reserved
3	*Reserved
4	Signal Ground (SGND)
5	Transmitted (TxD) (Out)
6	Received (RxD)(In)
7	NC
8	NC

External clock input port pinouts

The TAC cards provide an external clock input port to connect T1/E1 or BITS external clock reference.

[Table 239](#) lists the pinouts for the TAC card clock port. Pinouts follow the standard RJ45 specifications with pins 1 and 2 for receive and pins 4 and 5 for transmit. Pins 6, 7, and 8 are used for 2.048 MHz square wave signals when the line-type in the DS1 profile is set to *other*.

* Connect BITS select to ground to use BITS clock input.

Table 239: TAC card external clock pinouts

Pin	Function
1 T1/E1	Rx ring
2 T1/E1	Rx tip
3	Not used
4 T1/E1	Tx ring
5 T1/E1	Tx tip
6	BITS Select *

Table 239: TAC card external clock pinouts

Pin	Function
7	BITS clock
8	GND

19

SMALL FORM FACTOR PLUGGABLE (SFP) CONNECTORS

This chapter describes the Small Form Factor Pluggables (SFPs) and XFPs used by the MXK and covers:

- [Small form factor pluggables \(SFPs\), page 1579](#)
- [Insert and remove a fiber connection and an SFP, page 1583](#)
- [Insert and remove a dual bi-directional SFP and fiber connector, page 1584](#)
- [View SFP information on the MXK, page 1586](#)

Small form factor pluggables (SFPs)

Zhone Technologies supports a variety of small form factor pluggables (SFPs) and XFPs which are selected depending on the protocol, fiber type and distance requirements. This section covers:

- [SFPs for MXK uplink cards, page 1580](#)
- [XFPs for MXK uplink cards, page 1581](#)
- [SFPs for MXK Active Ethernet line cards, page 1581](#)
- [GPON SFP specifications, page 1582](#)

SFPs for 10 Gig ports on MXK uplink and Active Ethernet line cards

[Table 240](#) describes the SFPs for the 10 Gig Ethernet ports on the

- MXK-UPLINK-2X10G-8X1GE-CLK uplink card
- MXK-AE-2X10G-8X1GE line card

Table 240: SFPs for 10 Gig Ethernet ports

SFPs	Description
MXK-10GE-SFP+-SR	SFP+ SHORT REACH (300M), MULTI MODE, 850NM, DUPLEX LC/UPC, SUPPORTING 10GBPS ETHERNET; I-TEMP

Table 240: SFPs for 10 Gig Ethernet ports (Continued)

SFPs	Description
MXK-10GE-SFP+-20KM-1310	SFP+ LONG REACH (20KM), SINGLE MODE, 1310NM, DUPLEX LC/UPC, SUPPORTING 10GBPS ETHERNET; I-TEMP
MXK-10GE-SFP+-40KM-1550	SFP+ LONG REACH (40KM), SINGLE MODE, 1550NM, DUPLEX LC/UPC, SUPPORTING 10GBPS ETHERNET; I-TEMP
MXK-10GE-SFP+-80KM-1550	SFP+ LONG REACH (80KM), SINGLE MODE, 1550NM, DUPLEX LC/UPC, SUPPORTING 10GBPS ETHERNET; I-TEMP

SFPs for 1 GE ports

[Table 241](#) describes the SFPs for the 1GE Ethernet ports on the

- MXK-UPLINK-2X10G-8X1GE-CLK uplink card
- MXK-AE-2X10G-8X1GE line card

Table 241: SFPs for 1 GE ports

SFPs	Description
SFP-GE-SX-850-DLC	SFP GE SX (1000MBPS) TX 850 NM RX 850 NM UP TO 500M W/ DUPLEX LC CONNECTOR IND TEMP
SFP-GE-LX-1310-DLC	SFP GE LX (1000MBPS) TX 1310 NM RX 1310 NM UP TO 10KM W/ DUPLEX LC CONNECTOR IND TEMP
SFP-GE-EX-1310-DLC	SFP GE EX (1000MBPS) TX 1310 NM RX 1310 NM UP TO 40KM W/ DUPLEX LC CONNECTOR IND TEMP
SFP-GE-ZX-1550-DLC	SFP GE ZX (1000MBPS) TX 1550 NM RX 1550 NM UP TO 80KM W/ DUPLEX LC CONNECTOR IND TEMP
SFP-GE-BX-1310-SLC	SFP GE BX (1000MBPS) TX 1310 NM RX 1490 NM UP TO 10KM W/ SIMPLEX LC CONNECTOR IND TEMP
SFP-GE-BX-1490-SLC	SFP GE BX (1000MBPS) TX 1490 NM RX 1310 NM UP TO 10KM W/ SIMPLEX LC CONNECTOR IND TEMP
SFP-GE-BEX-1310-SLC	SFP GE BEX (1000MBPS) TX 1310 NM RX 1550 NM UP TO 40KM W/ SIMPLEX LC CONNECTOR IND TEMP
SFP-GE-BEX-1550-SLC	SFP GE BEX (1000MBPS) TX 1550 NM RX 1310 NM UP TO 40KM W/ SIMPLEX LC CONNECTOR IND TEMP

SFPs for MXK uplink cards

MXK uplink cards support four or more Gigabit Ethernet ports that connect the MXK to the network. The MXK uplink cards use pluggable optics for maximum flexibility. Zhone provides a variety of Gigabit SFPs which are tested and verified to work in the MXK.

The part numbers for these SFPs begin with SFP-GE-.

XFPs for MXK uplink cards

The XFP (10 Gigabit Small Form Factor Pluggable) is the pluggable transceiver used on the 10 Gigabit ports on the MXK uplink cards. Zhone provides several XFP's which operate over various distances.

These XFP parts all begin with MXK-10GE-XFP-.

SFPs for MXK Active Ethernet line cards

The Active Ethernet line cards for the MXK use pluggable optics for flexibility and the ability to add additional optics as the network grows.

Single-channel SFPs

Single-channel SFPs are SFPs that support a single subscriber and may use one or two fibers to transmit and receive wavelengths.

MXK line cards supporting single-channel SFPs:

- MXK-AEX20-FE/GE-2S
- MXK-AEX20-FE/GE-CSFP

Zhone's single-channel SFPs are available in both Fast Ethernet and Gigabit Ethernet speeds.

Part numbers for single channel SFPs begin with:

- SFP-FE-
- SFP-GE-

Dual-channel SFPs

Dual-channel SFPs are SFPs that support two subscribers. Dual-channel SFPs use two fibers with each fiber carrying both the transmit and receive wavelengths to the subscriber.

The dual-channel SFPs with the part number prefix MXK-AE-SFP-DL-BIDI- is only supported in the line card MXK-AEX-20-FE/GE and is the only SFP that the MXK-AEX-20-FE/GE line card supports.

The dual-channel SFPs with the part number prefix CSFP-GE- are supported in the line card MXK-AEX20-FE/GE-CSFP. This line card also supports single channel SFPs.



Note: The subscriber side connection to the SFP should have the opposite transmit and receive frequency. If a 1310 nm Transmit, 1550 nm Receive SFP is used on the single slot Active Ethernet card, the other side must have a 1310 Receive and 1550 Transmit.

[Table 242](#) describes the Active Ethernet line cards on the MXK and which SFPs they support.

Table 242: SFP support for Active Ethernet line cards

Card	Supports SFPs that begin with
MXK-AEX20-FE/GE	MXK-AE-SFP-DL-BIDI
MXK-AEX20-FE/GE-2S	SFP-FE SFP-GE
MXK-AEX20-FE/GE-CSFP	SFP-FE SFP-GE CSFP-GE

GPON SFP specifications

The SFP simple SC connector is a Burst receive GPON OLT transceiver with the specifications described in [Table 243](#).

Table 243: SFP specifications

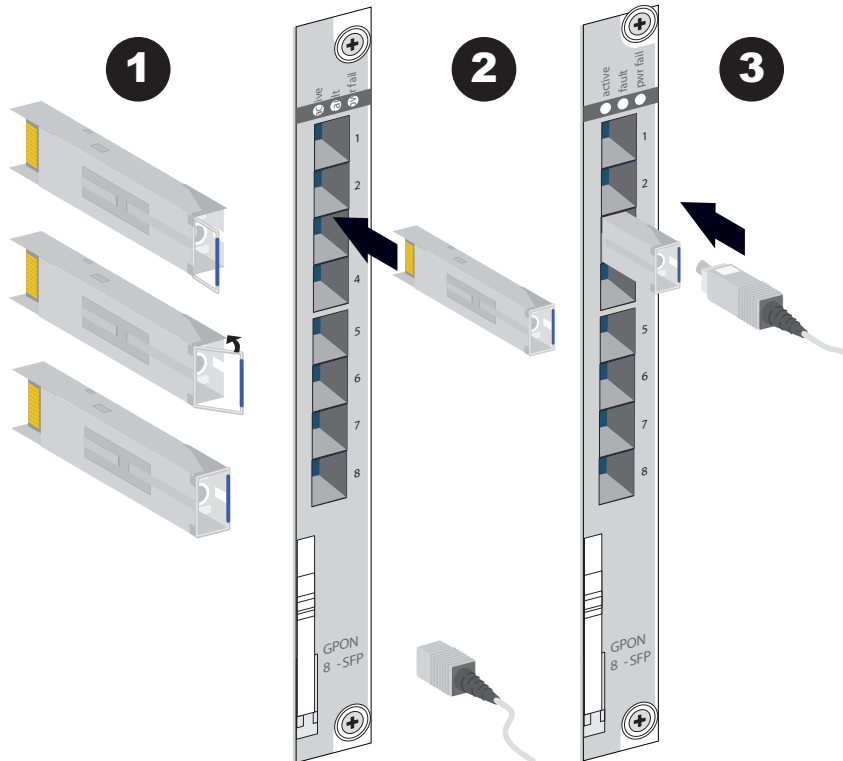
Class B+ Optics	Class C+ Optics
20 km reach; -28 dB link budget	60 km maximum reach; -32 dB link budget
“Fast Signal Detect” feature reduces ranging overhead	“Fast Signal Detect” feature reduces ranging overhead
Simplified OLT “reset” timing	Simplified OLT “reset” timing
1490 nm Transmit wavelength	1490 nm Transmit wavelength
1310 nm Receive Wavelength	1310 nm Receive Wavelength
2488 Mbps downstream	2488 Mbps downstream
1244 Mbps upstream Rx	1244 Mbps upstream Rx
Single 3.3 V supply	Single 3.3 V supply
ITU-T G.984.2 compliant	ITU-T G.984.2 compliant
RoHS-5/6 compliant (lead exemption)	RoHS-5/6 compliant (lead exemption)
RSSI and DDM (compliant with SFF8472 rev.9.5) supported	RSSI and DDM (compliant with SFF8472 rev.9.5) supported
operating and storage temperature -40 to +85C	operating and storage temperature -40 to +85C
optical power 1.5W	optical power 1.6W

Insert and remove a fiber connection and an SFP

Zhone fiber connections use Small Form Factor Pluggable (SFP) connectors. These connectors may be used with 10 Gigabit Ethernet, Gigabit Ethernet, Fast Ethernet/Gigabit Ethernet, Active Ethernet, and GPON cards.

Inserting a fiber connection and an SFP

- 1 On the SFP, push the handle wire latch in.



- 2 Slide the SFP in the port.
You should hear a slight click. Note that the SFP is not flush with the face of the card.
- 3 Insert the fiber connector into the SFP.

Removing a fiber connection and SFP

Removing SFP connectors from 10 Gigabit Ethernet, Gigabit Ethernet, Fast Ethernet/Gigabit Ethernet, Active Ethernet, or GPON cards is the reverse of installation.

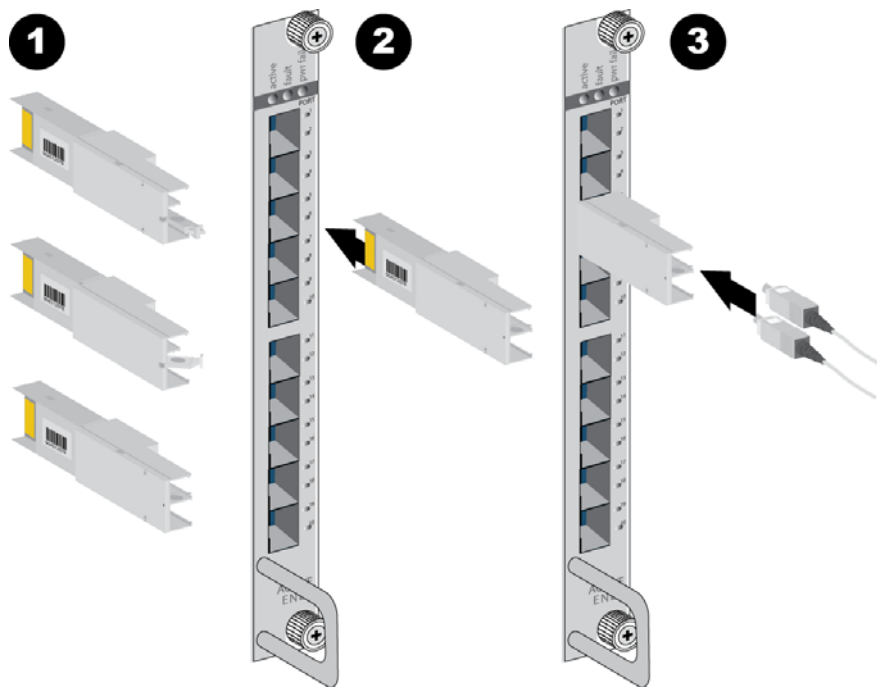
- 1 Remove the fiber connector from the SFP.
- 2 On the SFP, pull the handle wire latch outward.
- 3 Pull the SFP from the port.

Insert and remove a dual bi-directional SFP and fiber connector

Inserting a dual bi-directional SFP and fiber connector

The single slot 20 port Active Ethernet card uses dual bi-directional SFP connectors. Each physical port on the card is split into two physical ports on the SFP into which you can insert a fiber connection. Each fiber connection supports traffic in both directions.

- 1 Push in the handle latch which is between the two ports on the dual bi-directional SFP.



- 2 Slide the SFP in the port.
You should hear a slight click.



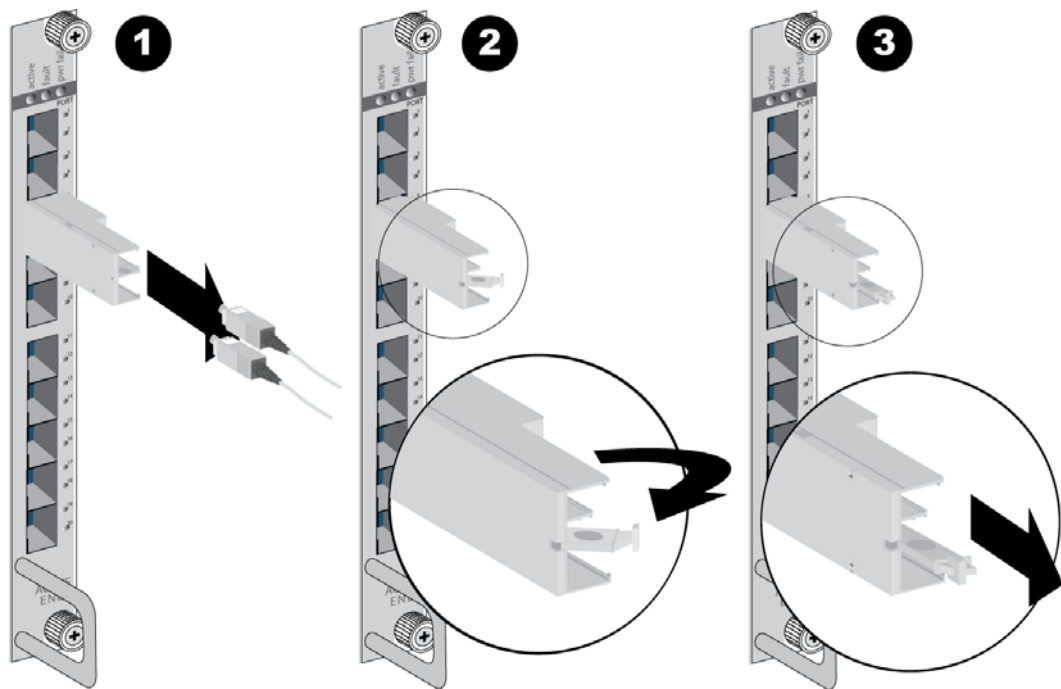
Note: The SFP is not flush with the face of the card.

- 3 Insert the fiber connector(s) into the SFP.

Removing the fiber connections and dual bi-directional SFP

Removing SFP connectors from single slot Active Ethernet cards is the reverse of installation.

- 1 Remove the fiber connector(s) from the SFP.



- 2 Pull the handle latch outward on the SFP.
- 3 Pull the SFP from the port.

View SFP information on the MXK

To view the presence of SFPs on the MXK, enter the **sfp show all** command:

```
zSH> sfp show all
SFP Data for interface 1-a-4-0/eth
vendorName                FINISAR CORP.
vendorOui                 00-90-65
vendorPartNumber          FCLF-8521-3
vendorRevisionLevel       A
serialNumber              PDJ3FGE
manufacturingDateCode     080504
complianceCode            base1000T (0x0008)
connectorType             unknownOrUnspecified (0)
transceiverType           sfp (3)
extendedIdentifier        4
encodingAlgorithm         eightb10b (1)
channelLinkLength         unknown value (0x0000)
channelTransmitterTechnology unknown value (0x0000)
channelTransmitterMedia   unknown value (0x0000)
channelSpeed              unknown value (0x0000)
nineTo125mmFiberLinkLengthKm 0
nineTo125mmFiberLinkLength100m 0
fiftyTo125mmFiberLinkLength10m 0
sixtyTwoDot5To125mmFiberLinkLength10m 0
nominalBitRate            12
upperBitRateMarginPercentage 0
lowerBitRateMarginPercentage 0
copperLinkLength          100
SFP Data for interface 1-a-5-0/eth
vendorName                FINISAR CORP.
vendorOui                 00-90-65
vendorPartNumber          FCLF-8521-3
vendorRevisionLevel       A
serialNumber              PD43QGK
manufacturingDateCode     080125
complianceCode            base1000T (0x0008)
connectorType             unknownOrUnspecified (0)
transceiverType           sfp (3)
extendedIdentifier        4
encodingAlgorithm         eightb10b (1)
channelLinkLength         unknown value (0x0000)
channelTransmitterTechnology unknown value (0x0000)
channelTransmitterMedia   unknown value (0x0000)
channelSpeed              unknown value (0x0000)
nineTo125mmFiberLinkLengthKm 0
nineTo125mmFiberLinkLength100m 0
fiftyTo125mmFiberLinkLength10m 0
sixtyTwoDot5To125mmFiberLinkLength10m 0
nominalBitRate            12
upperBitRateMarginPercentage 0
lowerBitRateMarginPercentage 0
copperLinkLength          100
SFP Data for interface 1-a-6-0/eth
```

```

** No SFP present **
SFP Data for interface 1-a-7-0/eth
** No SFP present **
SFP Data for interface 1-a-8-0/eth
** No SFP present **
SFP Data for interface 1-a-9-0/eth
** No SFP present **
SFP Data for interface 1-a-10-0/eth
** No SFP present **
SFP Data for interface 1-a-11-0/eth
** No SFP present **
SFP Data for interface 1-4-1-0/gponolt
vendorName                LUMINENTOIC
vendorOui                 00-06-b5
vendorPartNumber          SPS4348HHPRDE
vendorRevisionLevel       1
serialNumber              8bma100050
manufacturingDateCode     081023
complianceCode            unknown value (0x0000)
connectorType             sc (1)
transceiverType           sfp (3)
extendedIdentifier        4
encodingAlgorithm          nrz (3)
channelLinkLength         unknown value (0x0000)
channelTransmitterTechnology unknown value (0x0000)
channelTransmitterMedia   unknown value (0x0000)
channelSpeed              unknown value (0x0000)
nineTo125mmFiberLinkLengthKm 20
nineTo125mmFiberLinkLength100m 200
fiftyTo125mmFiberLinkLength10m 0
sixtyTwoDot5To125mmFiberLinkLength10m 0
nominalBitRate            25
upperBitRateMarginPercentage 0
lowerBitRateMarginPercentage 0
copperLinkLength          0
SFP Data for interface 1-4-2-0/gponolt
** No SFP present **
SFP Data for interface 1-4-3-0/gponolt
** No SFP present **
SFP Data for interface 1-4-4-0/gponolt
** No SFP present **
SFP Data for interface 1-5-1-0/gponolt
vendorName                LUMINENTOIC
vendorOui                 00-06-b5
vendorPartNumber          SPS4348HHPRDE
vendorRevisionLevel       1
serialNumber              8bma100146
manufacturingDateCode     081023
complianceCode            unknown value (0x0000)
connectorType             sc (1)
transceiverType           sfp (3)
extendedIdentifier        4
encodingAlgorithm          nrz (3)
channelLinkLength         unknown value (0x0000)

```

Small Form Factor Pluggable (SFP) Connectors

channelTransmitterTechnology	unknown value (0x0000)
channelTransmitterMedia	unknown value (0x0000)
channelSpeed	unknown value (0x0000)
nineTo125mmFiberLinkLengthKm	20
nineTo125mmFiberLinkLength100m	200
fiftyTo125mmFiberLinkLength10m	0
sixtyTwoDot5To125mmFiberLinkLength10m	0
nominalBitRate	25
upperBitRateMarginPercentage	0
lowerBitRateMarginPercentage	0
copperLinkLength	0
SFP Data for interface 1-5-2-0/gponolt	
** No SFP present **	
SFP Data for interface 1-5-3-0/gponolt	
** No SFP present **	
SFP Data for interface 1-5-4-0/gponolt	
** No SFP present **	
SFP Data for interface 1-5-5-0/gponolt	
** No SFP present **	
SFP Data for interface 1-5-6-0/gponolt	
** No SFP present **	
SFP Data for interface 1-5-7-0/gponolt	
** No SFP present **	
SFP Data for interface 1-5-8-0/gponolt	
** No SFP present **	
SFP Data for interface 1-6-1-0/eth	
vendorName	OptoMedia
vendorOui	00-00-00
vendorPartNumber	UB4-S4-4103L-A
vendorRevisionLevel	
serialNumber	0070850040
manufacturingDateCode	100617
complianceCode	base1000Lx (0x0002)
connectorType	1c (7)
transceiverType	sfp (3)
extendedIdentifier	4
encodingAlgorithm	eightb10b (1)
channelLinkLength	unknown value (0x0000)
channelTransmitterTechnology	unknown value (0x0000)
channelTransmitterMedia	unknown value (0x0000)
channelSpeed	unknown value (0x0000)
nineTo125mmFiberLinkLengthKm	10
nineTo125mmFiberLinkLength100m	100
fiftyTo125mmFiberLinkLength10m	0
sixtyTwoDot5To125mmFiberLinkLength10m	0
nominalBitRate	12
upperBitRateMarginPercentage	0
lowerBitRateMarginPercentage	0
copperLinkLength	0
SFP Data for interface 1-6-2-0/eth	
** No SFP present **	
SFP Data for interface 1-6-3-0/eth	
** No SFP present **	
SFP Data for interface 1-6-4-0/eth	


```
** No SFP present **  
SFP Data for interface 1-6-5-0/eth  
** No SFP present **  
SFP Data for interface 1-6-6-0/eth  
** No SFP present **  
SFP Data for interface 1-6-7-0/eth  
** No SFP present **  
SFP Data for interface 1-6-8-0/eth  
** No SFP present **  
SFP Data for interface 1-6-9-0/eth  
** No SFP present **  
SFP Data for interface 1-6-10-0/eth  
** No SFP present **  
SFP Data for interface 1-6-11-0/eth  
** No SFP present **  
SFP Data for interface 1-6-12-0/eth  
** No SFP present **  
SFP Data for interface 1-6-13-0/eth  
** No SFP present **  
SFP Data for interface 1-6-14-0/eth  
** No SFP present **  
SFP Data for interface 1-6-15-0/eth  
** No SFP present **  
SFP Data for interface 1-6-16-0/eth  
** No SFP present **  
SFP Data for interface 1-6-17-0/eth  
** No SFP present **  
SFP Data for interface 1-6-18-0/eth  
** No SFP present **  
SFP Data for interface 1-6-19-0/eth  
** No SFP present **  
SFP Data for interface 1-6-20-0/eth  
** No SFP present **
```


INDEX

Numerics

- 10 GE or 100/1000 Ethernet (in-band) 44
- 10/100 Base T Ethernet port (out-of-band) 44, 48
- 48-port ADSL+POTS cards 677
- 802.1 Q-in-Q (VLAN tagging) 254
- 802.3ad link aggregation 589
- 802.3ah EFM OAM 1463
- 802.3ah EFM standards 1411

A

- acronym definitions 29
- activating slot cards
 - slot card installation 1531
- Active Ethernet line card dual-slot 1131
 - additional card information 1135
 - card type 1133, 1148
 - configuration 1133, 1148
 - overview 1132
 - specifications 1133
- Active Ethernet line card single-slot
 - additional card information 1140
 - card type 1138
 - configuration 1138
 - overview 1137
 - specifications 1138
- Active Ethernet line card single-slot with C-SFP
 - additional card information 1145
 - card type 1143
 - configuration 1143
 - overview 1142
 - specifications 1143
- adding a user 68
- administration
 - deleting user account 71
 - port administration 111
 - user accounts 68
- ADSL configurable options 1202
- ADSL interfaces
 - verifying the interface 1259, 1283
- ADSL overview 1202
- ADSL2+ bond card
 - ADSL overview 1202
 - ADSL2+ bonding 1272, 1276
 - adsl-co-profile parameters 1218
 - adsl-cpe-profile parameters 1227
 - adsl-profile defaults 1214
 - adsl-profile parameters 1215
- ATM data connection 1279
- ATM service ranges 1279
- ATM traffic descriptors 1281
- Broadcom Phy-R parameters 1255
- cable and port pinouts 1299
- card types 1190, 1193, 1199
- card-profile 1190, 1198
- configuration 1190, 1193, 1198, 1213
- configure ADSL2+ S=1/2 1249
- configure Annex M 1235
- configure capping train rates 1244
- configure G.lite 1241
- configure upstream and downstream tone ranges 1234
- create a gbond group 1272
- DSL statistics 1259, 1283
- fast configuration 1210
- interleaved configuration 1211
- MEGACO configuration 511
- MGCP configuration 507
- overview 1185, 1186
- profiles, adsl-profile, adsl-co-profile, adsl-cpe-profile 1213
- rate adaption 1203
- Seamless Rate Adaptation 1207
- SELT and DELT testing 1334
- signal-to-noise (SNR) parameter 1204
- SIP dial plan configuration 485
- SIP PLAR server configuration 500
- SIP server configuration 483
- SNR performance 1206
- specifications 1187, 1197
- transmission modes 1203
- transport mode, fast or interleaved 1209
- view additional card information 1195, 1200
- VoIP overview 481
- VPI and VCI ranges 1279
- ADSL2+ bond cards
 - ADSL configurable options 1202
- ADSL2+ bonding 1272, 1276
- ADSL2+ bonding for ADSL
 - bridging and routing 1085
 - bridging on ADSL2+ 1086, 1095
- ADSL2+ fallback on VDSL2
 - overview 1065
- ADSL2+ POTS line card
 - 48-port ADSL2+ POTS card configuration 1363
 - card types 1363
 - card-profile 1363

- ADSL2+ SNR 1204
- ADSL2+ transmission modes 1203
- alarms 139, 1569
 - external on MTAC/Ring cards 1569
- A-Law
 - setting 469
- Annex M 1235
- ATM
 - statistics 1281
 - traffic descriptor configuration rules 1280
 - traffic descriptors 1280
- ATM data connection 1279
 - traffic descriptors 1280
- ATM data connection configuration 1279
- ATM service ranges 1279
- ATM traffic descriptors 1281
- auto-negotiate or specific data rate for G. SHDSL 1425

B

- bandwidth limiting by port and service 206, 334
- baud rate adaption for G.SHDSL 1424
- BER test description 1502
- bond group bandwidth for EFM SHDSL 1411
- bond group creation 1414
- bond group overview for EFM 1414
- bonded copper pairs 1410
- BRAs 378
- bridge loop detection alarms 280
- bridge loop prevention 275
- bridge types
 - asymmetric intralink bridge 222
 - asymmetric uplink bridge 222
 - asymmetric downlink bridge 222
 - symmetric TLS bridges 217
 - symmetric wire bridges 217
- bridged video
 - bridged video with MVR 443, 456
 - configuration 429
 - IGMP snooping with proxy reporting
 - custom IP address 436
 - multicast control lists 430
 - VLAN translation 440
 - VLAN translation and MVR 439, 447
- bridging
 - add filters 313
 - bridge add command 207
- bridging commands 413, 424
- bridging concepts 201
 - bridge add command 201
 - bridge interfaces 205
 - broadcast frames 209
 - logical interface 205
 - multicast 209
 - physical interface 204
 - physical port 204
 - Quality of Service (QoS) for traffic separation 206
 - tagged frame (double tagged) 207
 - tagged frame (single tagged) 207
 - terminology 203
 - TLS bridges
 - floodUnknown and floodMulticast 247
- traffic separation
 - bandwidth limiting by port and service 206
 - destination MAC swapping 206
 - DHCP relay 206
 - forbid OUI 206
 - PPPoE with intermediate agent 206
- unicast frames 209
- untagged frames 207
- upstream and downstream 208
- VLAN separation of traffic 205
- VLANs and SLANs 205
- bridging configuration
 - bridge loop detection alarms 280
 - bridge loop prevention 275
 - bridging configurations 239
 - downlink bridges
 - tagged on Active Ethernet 245
 - tagged or untagged with VLAN ID 243
 - untagged on Active Ethernet 244
 - dynamic IP filtering on a bridge (secure DHCP)
 - secure DHCP 283
 - intralink bridges 227
 - tagged with VLAN 0 270
 - tagged with VLAN 0 268
- Q-in-Q
 - VLAN ID and SLAN ID 254
- TLS bridges 220
 - bridge-path defaults 221, 233
 - with VLAN 0 266
- TLS bridges configuraton 246
- uplink and downlink bridges
 - on GPON for triple-play 294
 - with VLAN 0 264
- uplink bridges
 - tagged with VLAN ID 239

- VLAN O (VLAN wildcard) 263
- VLAN translation 298
 - asymmetrical bridges 301
 - rules 298
 - TLS bridges 299
 - with SLAN promotion on asymmetrical bridges 303
- broadcast frames 209
- broadcast storm protection 355

C

- cable pinouts for ADSL2+ 1299
- cables
 - EFM T1/E1-24 card cable 1514
- cables for EFM T1/E1 1493
- call conferencing, SIP 525
- call progress parameters 473
- caller-id-sig-protocol 474
- capping upstream and downstream train rates 1244
- card configuration
 - Active Ethernet dual-slot 1133
 - Active Ethernet single-slot 1138
 - Active Ethernet single-slot with C-SFP 1143
 - EFM SHDSL 24-port line cards 1406
 - Ethernet uplink cards 607
 - POTS 72-port line card 1359
 - PWE T1/E1 24-port line card 1511
 - VDSL2 line cards 1016, 1023
- card information 148
- card types
 - ADSL2+ bond card 1190, 1193, 1199
- cards
 - types 1530
 - viewing active redundant 1532
- change default passwords, how to 70
- chassis information 73
- circuit emulation service (CES) 1510
- Class of Service (COS) 407
- class of service queuing 406
- clid-mode 475
- clocking
 - defaults 174
 - external clock on MTAC/Ring 1562
 - set line card as network timing source 180
 - set uplink and line card as source 180
 - set uplink card as network timing source 180
 - set uplink card as source 181
 - sources for the system 177

- clocking source
 - EFM SHDSL switch clocking source 1409
- color aware rate limiting 347
- color blind rate limiting 337
- concurrent management sessions 48
- configurable jitter buffer 477
- configuration
 - verifying interfaces 1259, 1283
- configuring physical interfaces
 - verifying interfaces 1259, 1283
- connect LP card to CPE devices 1467
- Constant Bit Rate (CBR) 1279
- constellation settings for G.SHDSL 1426
- COS, in VLAN headers 407
- CPE profile
 - copy
 - CPE profile
 - move 919, 921
- craft port settings 48
- current condition minimum threshold SNR margin 1431
- current condition SNR maximum threshold 1431

D

- default login 45
- default password 45
- default passwords, changing 70
- deleting a user 70
- deleting a user, description of 71
- DELT testing 1334
- destination MAC swapping 206, 352
- DHCP relay 206
- DHCP relay agent 324
- dialing plan 485
- DNS resolver 157
- downlink and uplink bridges
 - with VLAN 0 264
- downlink bridges
 - tagged on Active Ethernet 245
 - tagged or untagged with VLAN ID 243
 - untagged on Active Ethernet 244
- downloading software files 94
- downstream power backoff (DPBO) on VDSL2 1106
- ds1 interface activation 1477
- ds1-profile parameters 1478
- DSA 130
- DSL statistics on ADSL2+ bond cards 1259, 1283

dump command 105

E

EFM bond group overview 1414

EFM OAM 1463

EFM SHDSL 24-port line cards

auto-negotiate or specific data rate 1425

bond group bandwidth 1411

bond groups 1411

card configuration 1406

card type 1406

card-profile 1406

connecting LP card to CPE 1467

constellation settings 1426

create bond groups 1414

current condition minimum threshold SNR margin 1431

current condition SNR maximum threshold 1431

deliver power and data to the CPE 1467

EFM bond group overview 1414

error monitoring 1420

Ethernet bonding 1411

Ethernet services over SHDSL links 1410

network scenario with bonded copper pairs 1410

NTP and NTWC 1403

overview 1404

parameters to configure SNR monitoring 1431

pinouts 1465

pme-profile (Physical Medium Entities) 1423

port statistics 1456

regional settings 1428

set SNR for target current condition or target worst case mode 1434

set SNR monitoring from CLI 1435

set wetting current 1408

SHDSL error monitoring statistics 1449

SNR crossing traps 1442

SNR maintenance mode settings 1433

SNR monitoring for bonded lines 1430

SNR monitoring in the pme-profile 1439

SNR monitoring overview 1430

SNR monitoring statistics 1438

SNR pme-profile and efm-port parameters 1433

specifications 1405

switch clocking source 1409

TAC testing 1470

TCPAM configuration 1426

EFM T1/E1 24-port line card

activate a ds1 interface 1477

BER test description 1502

cables 1493

card-profile 1474

configuration 1474

ds1-profile parameters 1478

N2N bonding 1487

overview 1472

port statistics 1488

specifications 1473

Emergency Stand Alone (ESA) SIP support 489, 501

error monitoring for EFM SHDSL 1420

ESA support 489, 501

Ethernet

enhanced port statistics 1168

interfaces 1152

Ethernet interfaces 668

Ethernet OAM 1463

Ethernet services over SHDSL links 1410

Ethernet uplink cards 607

overview 607

pinouts 621, 670

specifications 609

F

fan tray monitoring 74

fax service, T.38 545

file system navigation 93

fix-bit rate settings 1425

floodMulticast 247

floodUnknown 247

forbid OUI 206, 323

G

G.SHDSL automatic baud rate adaption 1424

gain settings 1370

gbond groups 1272

Generic profile

creation 702

definition 694

deletion 717

import/export 720, 919, 921

GPON bridges for triple-play 294

- GPON card
 - CPE profile 919, 921
 - extended reach 965
 - Generic profile 694, 702, 717, 720, 919, 921
 - ME profile 694, 702, 717, 720
 - OMCI overview 693
 - Smart OMCI configuration 697
 - Smart OMCI overview 693
 - Specific profile 694, 717, 720, 919, 921
- GPON card overview 675
- GPON card profiles 677
- GPON card specifications 676
- groundstart, configuring 479

H

- hookflash
 - configuring timers 523
- HTTP 128
- HTTPS (HTTP secure) 128

I

- IGMP join and leave requests 428
- IGMP proxy on bridged video 428
- IGMP snooping with proxy reporting
 - custom IP address 436
- interfaces
 - specifying type of MTAC/Ring card 1530
- internal line testing 1375
- internetworking, PPPoA-PPPoE 378
- intralink
 - configuring bridges 227
- intralink bridges
 - staged with VLAN 0 270
 - tagged with VLAN 0 268
- IP and data support 39
- IP on a bridge 49

J

- jitter buffer 477

L

- LACP 589
 - available physical ports 591
 - enable Ethernet ports 595
- LEDs

- redundancy 620
- link aggregation 589
- Link Aggregation Control Protocol (LACP) 589
- log in and log out 45
- log messages 68
- log serial command 68
- log session command 68
- logging message format 83
- logging messages for the system 82
- logging on the MXK 81
- loopback configuration for T1/E1 1512
- loopstart, configuring 479

M

- map subscriber information to a port description
 - field 117
- ME profile
 - creation 702
 - definition 694
 - deletion 717
 - import 717
 - import/export 720
- MEGACO configuration for ADSL2+ 511
- metallic loop testing 1527
- MGCP configuration for ADSL2+ 507
- MKK features
 - rate limiting 40
- modem train rates 1425
- monitoring through serial craft port 68
- MTAC cards
 - internal look out test 1527
 - metallic loop testing 1527
 - ring generator 1528
 - specifications 1525
- MTAC/Ring card
 - configuring redundancy 1530
 - external alarm contacts 1569
 - specifying line type for 1530
- MTAC/Ring external contacts 1569
- Mu-Law
 - setting 469
- multicast control lists 430
- multicast frames 209
- MVR
 - bridged video 439, 443, 456
 - VLAN translation 447
- MX(P)-160/260 configuration
 - log in and log out 45

- MXK
 - IP and data support 39
- MXK card configuration
 - slot card provisioning 148
 - slots command
 - viewing card information 148
- MXK features
 - Megaco H.248 41
 - MGCP 41
 - SIP 41
 - VoIP 40
- MXK file system
 - commands 93
 - downloading software files 94
 - navigating the file system 93
- MXK line cards
 - EFM T1/E1-24 card 36
 - MXK-ADSL2+-BCM-48A 36
 - MXK-ADSL2+-POTS-BCM-48-2S 37
 - MXK-ADSL2+-POTS-BCM-48A-2S 36
 - MXK-ADSL2+-POTS-BCM-48A-RNG-2S 37
 - MXK-ADSL2+-SPLTR600-BCM-48A-2S 36
 - MXK-ADSL2+-SPLTR900-BCM-48A-2S 36
 - MXK-AEX20-FE/GE 35
 - MXK-AEX20-FE/GE-2S 35
 - MXK-BDSL2 BCM-17A-48-V 37
 - MXK-EFM-SHDSL-24 NTP 36
 - MXK-EFM-SHDSL-24-NTWC 36
 - MXK-GPONX4-IO 35
 - MXK-GPONX8-IO 35
 - MXK-MTAC/RING 38
 - MXK-MXK-MTAC/RNG-ENH 38
 - MXK-POTS-72 38
 - MXK-POTS-EBS-PKT-24 38
 - MXK-PWE-T1/E1-24 37
 - MXK-VDSL2-24-BCM 36
 - MXK-VDSL2-BCM-17A-48 37
 - MXK-VDSL2-POTS-BCM-17A-24 37
 - MXK-VDSL2-SPLTR600-BCM-17A-24 37
 - MXK-VDSL2-SPLTR900-BCM-17A-24 37
 - overview 35
- MXK management
 - 10 GE or 100/1000 Ethernet (in-band) 44
 - 10/100 Base T Ethernet interface (out-of-band) 48
 - 10/100 Base T Ethernet port (out-of-band) 44
 - access and manage from CLI 44
 - available ports 44
 - change port setting on craft port 48
 - concurrent management sessions 48
 - in-band (VLAN tagged) 39
 - initial system configuration 47
 - IP on a bridge 39, 49
 - out-of-band on the 10/100 Ethernet interface 39
 - serial craft RS 232 44, 47
 - VoIP on IP on a bridge for EAPS 57
 - Web UI 64
 - Zhone Management System (ZMS) 60
 - ZMS 59
- MXK network technologies 33
- MXK security
 - Digital Signature Algorithm (DSA) 130
 - features 128
 - HTTPS (HTTP secure) 128
 - port access security 132
 - RSA 130
 - Secure File Transfer Protocol (SFTP) 128
 - secure shell (SSH) 128
 - SSH clients 130
 - SSH, SFTP, HTTPS 128
- MXK synchronized network timing clocking
 - set line card as network timing source 180
 - set uplink card as network timing source 180
 - set uplink card as source 181
- MXK synchronized network timing source
 - clocking sources 177
 - set uplink and line card as source 180
- MXK system administration
 - alarms 139
 - DNS resolver 157
 - log messages 68
 - log serial command 68
 - log session command 68
 - monitoring through serial craft port 68
 - SNMP 108
 - subscriber management
 - map subscriber information to a port
 - description field 117
 - port description rules 117
 - system defaults 67
 - system login 45
 - system password 45
- MXK system clocking
 - default clocking 174
- MXK system configuration
 - back up configuration to a local file 106
 - back up configuration to the network 106
 - dump command 105
 - restore the system configuration 106
 - save and restore the system configuration 105

- MXK system monitoring
 - fan tray monitoring 74
 - logging message format 83
 - logging on the MXK 81
 - shelfctrl monitor command 74
 - system logging messages 82
 - view chassis and slot information 73
- MXK uplink cards
 - MXK-UPLINK-2X10GE-8X1GE 34
 - MXK-UPLINK-4X1GE 34
 - MXK-UPLINK-4X1GE-CU 34
 - MXK-UPLINK-8X1G 34

N

- N2N bonding for EFM T1/E1 1487
- network scenario with bonded copper pairs 1410
- non-real-time variable bit rate (nrt-VBR) 1279

O

- OMCI overview 693

P

- packet-rule-record
 - bandwidth limiting by port and service 334
 - broadcast storm protection 355
 - color aware rate limiting 347
 - color blind rate limiting 337
 - destination MAC swapping 352
 - DHCP relay and Forbid OUI 323
 - filterfirstencapsulationvlan 262
 - option 82 DHCP on bridge
 - (bridgeinsertoption82) 316
 - option 82 DHCP on bridge with macro defined
 - strings 319
 - option 82 DHCP on bridge without macro
 - defined strings 318
 - PPPoE with intermediate agent
 - (bridgeinsertpppoevendortag) 327
 - PPPoE with intermediate agent with macro
 - defined strings 331
 - PPPoE with intermediate agent without macro
 - defined strings 329
 - promotefirstencapsulationvlan 261
- passwords, changing default 70
- pinouts 1465
 - external alarm 1569

- pinouts for ports and cables
 - POTS 72-port line card 1378
- pinouts for VDSL2 line cards 1126
- pme-profile
 - setting link rates 1423
- pme-profile (Physical Medium Entities) 1423
- port access security 132
- port administration 111
- port command 111
- port description commands 118
- port description rules 117
- port mirroring 123
- port pinouts for ADSL2+ 1299
- port statistics for EFM T1/E1 1488
- POTS
 - configuring groundstate 479
 - configuring loopstart 479
- POTS 72-port line card
 - card type 1350
 - configuration 1358
 - internal line testing 1375
 - pinouts for ports and cables 1378
 - specifications 1348
- POTS analog-if-cfg-profile for gain settings 1370
- POTS card
 - 24 port card overview 1346
- POTS interface analog-fxs-cfg-profile for
 - signalling type and ring frequency 1373
- POTS interface configuration 1370
- power and data to the CPE 1467
- PPP tunnel 378
- PPPoA-PPPoE internetworking 378
- PPPoE with intermediate agent 206, 328
- provision slot cards 148
- PWE configuration
 - configuring PWE for E1 PRI 566
 - PWE solution with EAPS 561
 - PWE with CESoP channelization 563
 - T1/E1 loopback configuration 1512
- PWE on the MXK
 - bundles 552
 - cables 1514
 - CESoP packetization 558
 - latency issues with voice and data services 552
 - payload size and jitter buffer configuration 559
 - PWE UDP ports and IP addresses 557
 - PWE with T1 or E1 562
- PWE T1/E1 line card 1509
 - card-profile 1511
 - circuit emulation service (CES) 1510

- configuration 1511
- specifications 1511
- T1/E1 circuits 1510

Q

- Q-in-Q
 - staggered bridge 254
- Q-in-Q-in-Q 259
 - TLS bridges 261
- QoS and traffic descriptors
- QoS for traffic separation 206
- Quality of Service, see QoS

R

- RADIUS 134
- Rapid Spanning Tree Protocol (RSTP) 381
- rate limiting 40
- real-time variable bit rate (rt-VBR) 1280
- redundancy
 - LEDs 620
 - MTAC/Ring 1530
 - viewing active cards 1532
- regional settings for G.SHDSL 1428
- resetting passwords, description of 71
- restore the system configuration 106
- ring cadence 473
- ring generator on MTAC cards 1528
- RSA 130
- RSTP 381
 - on uplinks 383
 - port role 381
 - port state 382
 - rlinks 385

S

- Seamless Rate Adaptation 1207
- secure shell (SSH) 128
- security
 - Secure File Transfer Protocol (SFTP) 128
- SELT 1334
- SELT testing 1334
- serial craft RS232 44, 47
- server-max-timer, voice-system profile 483
- SFP 670
- SFP and XFP overview 1580
- SFPs 1154

- insert and remove a bi-directional SFP 1584
- inserting and removing an SFP 1583
- specifications 1582
- SFTP 128
- shaping traffic
 - class of service queuing 406
- SHDSL error monitoring statistics 1449
- SHDSL statistics 1452
- shelfctrl monitor command 74
- signalling type and ring frequency 1373
- Single End Loop Tests (SELT) 1334
- SIP dial plan configuration for ADSL2+ 485
- SIP PLAR server configuration for ADSL2+ 500
- SIP server configuration for ADSL2+ 483
- SIP, call conferencing 525
- SIP, calls not registering 483
- sip-dialplan 485
- slot cards
 - installation
 - verifying 1531
- slot information 73
- Small Form Factor Pluggables 670, 1154
- Small Form Factor Pluggables (SFPs) 1579
- Smart OMCI
 - Smart OMCI overview 693
 - Smart OMCI web-interface 697
- SNMP 108
- SNR 1204
- SNR crossing traps 1442
- SNR current condition maximum threshold 1431
- SNR current condition minimum threshold 1431
- SNR for target current condition or target worst case mode 1434
- SNR maintenance mode settings 1433
- SNR monitoring for bonded lines 1430
- SNR monitoring from CLI 1435
- SNR monitoring in the pme-profile 1439
- SNR monitoring overview 1430
- SNR monitoring parameters 1431
- SNR monitoring statistics 1438
- SNR performance 1206
- SNR pme-profile and efm-port parameters 1433
- Specific profile
 - definition 694
 - deletion 717
 - import/export 720, 919, 921
- SSH 128
- SSH clients 130
- staggered frame (double tagged) 207
- statistics

- ATM 1281
- statistics for SHDSL interfaces 1452
- statistics for SHDSL ports 1456
- statistics for VDSL2 1051, 1116
- subscriber management
 - port description commands 118
- system
 - data communications 1279
- system configuration save and restore 105
- system configuration, initial 47
- system defaults 67
- system profile
 - voice configuration 469

T

- T.38 fax service 545
- T1/E1 circuits 1510
- T1/E1 loopbacks, (for T1/E1 PWE and OC-3/STM-1 PWE connections) 1512
- T1/E1 PWE line card 1509
- TAC testing 1470
- tagged frame (single tagged) 207
- TCPAM configuration 1426
- TFTP server 724
- three-way call conferencing 525
- TLS bridges
 - bridge-path defaults 221, 233
 - configuration 220
 - wire bridges 217
 - with VLAN 0 266
- traffic descriptors
 - configuration rules 1280
 - description 1280
 - QoS
- training speeds for adsl-co-profile and adsl-cpe-profile 1205
- transmission rates for VDSL2 1027
- transport mode, fast or interleaved 1209
- types, listing of cards 1530

U

- ULC card
 - specifications 1346
- unicast frames 209
- unspecified bit rate (UBR) 1280
- untagged frames 207
- uplink and downlink bridges

- with VLAN 0 264
- uplink bridges
 - tagged with VLAN ID 239
- uplink card pinouts 621, 670
- uplink card specifications 609
- upstream power backoff (UPBO) on VDSL2 1104
- user accounts
 - adding a user 68
 - changing default passwords 70
 - deleting a user 70
 - deleting admin 71
 - resetting passwords 71

V

- VDSL2
 - downstream power backoff (DPBO) 1106
 - overview 1027
 - statistics 1051, 1116
 - transmission rates 1027
 - upstream power backoff (UPBO) 1104
 - VDSL2 standards 1027
 - VLSL2 on the MXK 1028
- VDSL2 bonding for VDSL
 - bridging 1099
 - bridging and routing 1089
- VDSL2 interfaces
 - profiles 1029
 - vdsl-co-config
 - Phy-r support 1034
 - vdsl-co-config parameters 1036
 - vdsl-co-config profile 1034
 - vdsl-config profile 1029
 - vdsl-config profile parameters 1030
 - vdsl-cpe-config
 - Phy-r support 1041
 - vdsl-cpe-config profile 1041
 - vdsl-cpe-config profile parameters 1043
- VDSL2 line cards
 - card types 1016, 1023
 - configuration 1016, 1023
 - overview 1013
 - pinouts 1126
 - specifications 1015
 - view additional card information 1019
- VDSL2+ POTS 24-port line card
 - card-profile 1367
 - configuration 1367
- vdsl-co-config

- Phy-r support 1034
- profile 1034
- profile parameters 1036
- vdsl-config
 - profile 1029
 - profile parameters 1030
- vdsl-cpe-config
 - Phy-r support 1041
 - profile 1041
 - profile parameters 1043
- view additional card information 1195, 1200
- VLAN 0 263
- VLAN separation of traffic 205
- VLAN translation 439
- VLAN translation 298, 440
 - asymmetrical bridges 301
 - rules 298
 - TLS bridges 299
 - with SLAN promotion on asymmetrical bridges 303
- VLAN wildcard 263, 268
- VLANs and SLANs 205
- voice
 - hookflash timers 523
 - POTS 24 card 1346
- Voice card
 - VoIP services 522
- voice configuration
 - system profile 469
- VOIP
 - call progress parameters 473
 - ring cadence 473
- VoIP
 - hookflash, configuring timers 523
- VoIP on IP on a bridge for EAPS 57
- VoIP overview for ADSL2+ 481
- VoIP services 522
- voip, country-specific dialing features 469
- VPI and VCI ranges 1279

Z

ZMS 59

W

- Web UI for MXK management 64
- wetting current 1408

X

- XFPs 1579