



7450 Ethernet Service Switch  
7750 Service Router  
7950 Extensible Routing System  
Virtualized Service Router  
Release 23.7.R1

## OAM and Diagnostics Guide

---

3HE 19231 AAAB TQZZA 01  
Edition 01  
July 2023

Nokia is committed to diversity and inclusion. We are continuously reviewing our customer documentation and consulting with standards bodies to ensure that terminology is inclusive and aligned with the industry. Our future customer documentation will be updated accordingly.

---

This document includes Nokia proprietary and confidential information, which may not be distributed or disclosed to any third parties without the prior written consent of Nokia.

This document is intended for use by Nokia's customers ("You"/"Your") in connection with a product purchased or licensed from any company within Nokia Group of Companies. Use this document as agreed. You agree to notify Nokia of any errors you may find in this document; however, should you elect to use this document for any purpose(s) for which it is not intended, You understand and warrant that any determinations You may make or actions You may take will be based upon Your independent judgment and analysis of the content of this document.

Nokia reserves the right to make changes to this document without notice. At all times, the controlling version is the one available on Nokia's site.

No part of this document may be modified.

NO WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF AVAILABILITY, ACCURACY, RELIABILITY, TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, IS MADE IN RELATION TO THE CONTENT OF THIS DOCUMENT. IN NO EVENT WILL NOKIA BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL OR ANY LOSSES, SUCH AS BUT NOT LIMITED TO LOSS OF PROFIT, REVENUE, BUSINESS INTERRUPTION, BUSINESS OPPORTUNITY OR DATA THAT MAY ARISE FROM THE USE OF THIS DOCUMENT OR THE INFORMATION IN IT, EVEN IN THE CASE OF ERRORS IN OR OMISSIONS FROM THIS DOCUMENT OR ITS CONTENT.

Copyright and trademark: Nokia is a registered trademark of Nokia Corporation. Other product names mentioned in this document may be trademarks of their respective owners.

© 2023 Nokia.

# Table of contents

<b>1</b>	<b>Getting started.....</b>	<b>11</b>
1.1	About this guide.....	11
1.2	Router configuration process.....	11
1.3	Conventions.....	12
1.3.1	Precautionary and information messages.....	12
1.3.2	Options or substeps in procedures and sequential workflows.....	12
<b>2</b>	<b>Mirror services.....</b>	<b>14</b>
2.1	Mirror implementation.....	14
2.1.1	Mirror components.....	15
2.1.2	Mirror source.....	15
2.1.2.1	Types and sources.....	16
2.1.2.2	Mirror source priority.....	17
2.1.3	Mirror destination.....	17
2.1.3.1	General local and remote mirroring.....	18
2.1.3.2	Mirror destination type IP-only.....	18
2.1.3.3	Layer 3 encapsulation.....	19
2.1.3.4	PCAP packet capture.....	21
2.1.3.5	Mirrored traffic transport using MPLS-TP SDPs.....	23
2.1.3.6	Slicing and sampling.....	30
2.1.4	Mirroring performance.....	31
2.2	Lawful Intercept.....	31
2.2.1	LI activation through RADIUS.....	32
2.2.2	Routable Lawful Intercept encapsulation.....	34
2.2.3	Lawful Intercept and NAT.....	37
2.2.3.1	Carrier grade NAT.....	37
2.2.3.2	L2-Aware NAT.....	38
2.2.4	Lawful Intercept management interfaces.....	39
2.2.4.1	LI management using the classic CLI engine.....	39
2.2.4.2	LI management using the MD-CLI engine.....	42
2.2.4.3	Lawful Intercept in NETCONF.....	43
2.2.4.4	Lawful Intercept in gNMI.....	44
2.2.4.5	Mixed mode SNMPv3 support.....	44

2.2.4.6	CLI configuration mode migration.....	44
2.2.5	Configuring Lawful Intercept in model-driven interface.....	48
2.3	Pseudowire redundant mirror services.....	49
2.3.1	Redundant mirror source notes.....	51
2.4	Configuration process overview.....	51
2.5	Configuration notes.....	53
2.6	Configuring service mirroring with CLI.....	54
2.6.1	Mirror configuration overview.....	55
2.6.1.1	Defining mirrored traffic.....	55
2.6.2	Lawful Intercept configuration overview.....	56
2.6.2.1	Saving LI data.....	56
2.6.2.2	Regulating LI access.....	56
2.6.2.3	Configurable filter lock for Lawful Intercept.....	60
2.6.2.4	LI MAC filter configuration.....	60
2.6.2.5	LI logging.....	61
2.6.3	Basic mirroring configuration.....	61
2.6.3.1	Mirror classification rules.....	62
2.6.4	Common configuration tasks.....	65
2.6.4.1	Configuring a local mirror service.....	65
2.6.4.2	Configuring SDPs for mirrors and LI.....	67
2.6.4.3	Configuring a remote mirror service.....	68
2.6.4.4	Configuring Lawful Intercept parameters.....	70
2.6.4.5	Pseudowire redundancy for mirror services configuration example.....	71
2.7	Service management tasks.....	72
2.7.1	Modifying a local mirrored service.....	72
2.7.2	Deleting a local mirrored service.....	73
2.7.3	Modifying a remote mirrored service.....	73
2.7.4	Deleting a remote mirrored service.....	74
<b>3</b>	<b>OAM fault and performance tools and protocols.....</b>	<b>76</b>
3.1	OAM overview.....	76
3.1.1	LSP diagnostics: LSP ping and LSP trace.....	76
3.1.2	LSP ping and LSP trace for an LSP using a BGP IPv4 or IPv6 label route.....	76
3.1.3	ECMP considerations.....	78
3.1.4	LSP ping and LSP trace over unnumbered IP interface.....	80
3.1.5	DDMAP TLV.....	80

3.1.6	Using DDMAP TLV in LSP stitching and LSP hierarchy.....	82
3.1.6.1	Responder node procedures.....	83
3.1.6.2	Sender node procedures.....	84
3.1.7	MPLS OAM support in Segment Routing.....	84
3.1.7.1	SR extensions for LSP ping and LSP trace.....	85
3.1.7.2	Operation on SR-ISIS or SR-OSPF tunnels.....	87
3.1.7.3	Operation on SR-TE LSP.....	88
3.1.7.4	Operation on an SR-ISIS tunnel stitched to an LDP FEC.....	91
3.1.7.5	Operation on a BGP IPv4 LSP resolved over an SR-ISIS IPv4 tunnel, SR-OSPF IPv4 tunnel, or SR-TE IPv4 LSP.....	92
3.1.7.6	Operation on an SR-ISIS IPv4 tunnel, IPv6 tunnel, or SR-OSPF IPv4 tunnel resolved over IGP IPv4 shortcuts using RSVP-TE LSPs.....	96
3.1.7.7	Operation on an LDP IPv4 FEC resolved over IGP IPv4 shortcuts using SR-TE LSPs.....	97
3.1.8	MPLS OAM support in IPv4 or IPv6 SR policies.....	100
3.1.8.1	LSP ping and LSP trace operation.....	101
3.1.8.2	ICMP-tunneling operation.....	103
3.1.9	Segment Routing IPv6 (SRv6) OAM.....	103
3.1.9.1	Ping or traceroute of SRv6 remote locator or remote SID (End, End.X, End.DT4, End.DT6, End.DT46, End.DX2, End.DT2M and End.DT2U ).....	105
3.1.10	LDP tree trace: end-to-end testing of paths in an LDP ECMP network.....	110
3.1.11	LDP ECMP tree building.....	111
3.1.12	Periodic path exercising.....	112
3.1.13	LSP ping for RSVP P2MP LSP (P2MP).....	112
3.1.14	LSP trace for RSVP P2MP LSP.....	113
3.1.14.1	LSP trace behavior when S2L path traverses a re-merge node.....	115
3.1.15	Tunneling of ICMP reply packets over MPLS LSP.....	116
3.1.16	QoS handling of tunneled ICMP reply packets.....	118
3.1.17	Summary of UDP traceroute behavior with and without ICMP tunneling.....	118
3.1.18	SDP diagnostics.....	119
3.1.19	SDP ping.....	119
3.1.20	SDP MTU path discovery.....	119
3.1.21	Service diagnostics.....	120
3.1.22	VPLS MAC diagnostics.....	120
3.1.23	MAC ping.....	121
3.1.24	MAC trace.....	121
3.1.25	CPE ping.....	122

3.1.26	CPE ping for PBB Epipe.....	122
3.1.26.1	Hardware support.....	123
3.1.27	MAC populate.....	123
3.1.28	MAC purge.....	124
3.1.29	VLL diagnostics.....	124
3.1.30	VCCV ping.....	124
3.1.30.1	VCCV-ping application.....	124
3.1.30.2	VCCV ping in a multi-segment pseudowire.....	126
3.1.31	Automated VCCV-trace capability for MS-pseudowire.....	127
3.1.31.1	VCCV for static pseudowire segments.....	128
3.1.31.2	Detailed VCCV-trace operation.....	128
3.1.31.3	Control plane processing of a VCCV echo message in a MS-pseudowire.....	128
3.1.32	IGMP snooping diagnostics.....	129
3.1.33	MFIB ping.....	129
3.1.34	MPLS-TP on-demand OAM commands.....	129
3.1.35	MPLS-TP pseudowires: VCCV ping/VCCV trace.....	130
3.1.35.1	VCCV ping and VCCV trace between static MPLS-TP and dynamic PW segments.....	130
3.1.36	MPLS-TP LSPs: LSP ping/LSP trace.....	131
3.1.37	VXLAN ping supporting EVPN for VXLAN.....	132
3.1.38	Show commands.....	133
3.1.39	BFD.....	133
3.1.40	MPLS Performance Monitoring (MPLS PM).....	134
3.1.40.1	Configuring MPLS PM.....	137
3.1.41	BIER OAM.....	142
3.1.41.1	ECMP and BIER OAM.....	142
3.1.41.2	Outbound time.....	142
3.1.41.3	Negative outbound time.....	142
3.1.42	ICMP ping check connectivity checking using ICMP echo request and response.....	143
3.2	IP Performance Monitoring (IP PM).....	146
3.2.1	TWAMP.....	147
3.2.2	TWAMP Light and STAMP.....	147
3.2.2.1	Overview.....	147
3.2.2.2	TWAMP Light Session-Reflector.....	149
3.3	MPLS PM.....	150
3.3.1	TWAMP Light delay and loss for MPLS tunnels.....	150

3.3.2	RFC 6374 delay for MPLS tunnels.....	151
3.4	ETH-CFM.....	154
3.4.1	ETH-CFM building blocks.....	156
3.4.2	Loopback.....	169
3.4.3	Loopback multicast.....	171
3.4.4	Linktrace.....	172
3.4.5	Continuity Check (CC).....	173
3.4.6	CC remote peer auto discovery.....	177
3.4.7	ETH-CFM grace overview.....	178
3.4.7.1	ETH-VSM grace (Nokia SR OS vendor-specific).....	179
3.4.7.2	ITU-T Y.1731 ETH-ED.....	180
3.4.8	CCM hold timers.....	180
3.4.9	ITU-T Y.1731 ETH-AIS.....	181
3.4.10	ITU-T Y.1731 ETH-CSF.....	183
3.4.11	ITU-T Y.1731 ETH-TST.....	183
3.4.12	ITU-T Y.1731 ETH-1DM.....	184
3.4.13	ITU-T Y.1731 ETH-DMM.....	184
3.4.14	ITU-T Y.1731 ETH-SLM.....	184
3.4.15	ITU-T Y.1731 ETH-LMM.....	186
3.4.15.1	ETH-LMM single SAP counter.....	188
3.4.15.2	ETH-LMM per forwarding class counter.....	189
3.4.15.3	Interaction between single and per FC counters.....	189
3.4.16	ETH-CFM destination options.....	190
3.4.17	ITU-T Y.1731 ETH-BN.....	191
3.4.18	ETH-CFM statistics.....	193
3.4.19	ETH-CFM packet debug.....	194
3.4.20	ETH-CFM CoS considerations.....	195
3.4.21	Silent CFM dropping with squelching.....	196
3.5	OAM mapping.....	198
3.5.1	CFM connectivity fault conditions.....	198
3.5.2	CFM fault propagation methods.....	199
3.5.3	Epipe services.....	200
3.5.4	CFM detected fault.....	200
3.5.4.1	SAP and SDP-binding failure (including pseudowire status).....	200
3.5.4.2	Service down.....	200
3.5.4.3	Interaction with pseudowire redundancy.....	201

3.5.5	lpipe services.....	201
3.5.5.1	SAP or SDP-binding failure (including pseudowire status).....	201
3.5.5.2	Service administratively shutdown.....	201
3.5.5.3	Interaction with pseudowire redundancy.....	201
3.5.6	VPLS service.....	201
3.5.6.1	CFM detected fault.....	201
3.5.6.2	SAP and SDP-binding failure (including pseudowire status).....	202
3.5.6.3	Service down.....	202
3.5.6.4	Pseudowire redundancy and Spanning Tree Protocol.....	202
3.5.7	IES and VPRN services.....	203
3.5.8	Pseudowire switching.....	203
3.5.9	LLF and CFM fault propagation.....	203
3.5.10	802.3ah EFM OAM mapping and interaction with service manager.....	203
3.6	Bidirectional Forwarding Detection.....	203
3.6.1	BFD control packet.....	204
3.6.2	Control packet format.....	204
3.6.3	Echo support.....	205
3.6.4	Centralized BFD.....	206
3.6.4.1	IES over spoke SDP.....	206
3.6.4.2	BFD over LAG and VSM interfaces.....	206
3.6.4.3	BFD on an unnumbered IPv4 interface.....	207
3.6.4.4	LSP BFD and VCCV BFD.....	207
3.6.4.5	Seamless BFD for SR-TE LSPs.....	208
3.6.5	Seamless Bidirectional Forwarding Detection.....	208
3.6.5.1	S-BFD reflector configuration and behavior.....	208
3.6.5.2	S-BFD initiator global configuration.....	209
3.7	Traceroute with ICMP tunneling in common applications.....	211
3.7.1	BGP-LDP stitching and ASBR/ABR/data path RR for BGP IPv4 label route.....	211
3.7.2	VPRN inter-AS option B.....	214
3.7.3	VPRN inter-AS option C and ASBR/ABR/data path RR for BGP IPv4 label route.....	215
3.8	Hashing visibility tool.....	217
3.8.1	Configuring the header templates.....	218
3.8.2	Configuring parameter overrides and header sequences.....	219
<b>4</b>	<b>OAM monitoring and reporting.....</b>	<b>220</b>
4.1	Link measurement.....	221



4.1.1	Link measurement template.....	222
4.1.1.1	General configuration.....	222
4.1.1.2	Collection and reporting.....	223
4.1.1.3	Protocol.....	224
4.1.1.4	Modifying measurement template configuration.....	226
4.1.1.5	Displaying link measurements.....	226
4.1.2	Interface assignment.....	226
4.1.2.1	IP addressing.....	226
4.1.2.2	Test initialization.....	227
4.1.2.3	History and results.....	228
4.1.2.4	Static versus dynamic.....	229
4.1.2.5	Displaying interface delay tests.....	229
4.2	OAM performance monitoring.....	229
4.2.1	Session.....	231
4.2.2	Standard PM packets.....	232
4.2.3	Measurement intervals.....	232
4.2.4	Data structures and storage.....	243
4.2.5	Bin groups.....	245
4.2.6	Delay results streaming.....	246
4.2.7	Relating the components.....	247
4.2.8	Monitoring.....	247
4.2.8.1	Accounting policy configuration.....	247
4.2.8.2	ETH-CFM configuration.....	248
4.2.8.3	Service configuration.....	248
4.2.8.4	OAM-PM configuration.....	248
4.2.8.5	Show and monitor commands.....	250
4.3	Service Assurance Agent.....	255
<b>5</b>	<b>Standards and protocol support.....</b>	<b>258</b>
5.1	Access Node Control Protocol (ANCP).....	258
5.2	Bidirectional Forwarding Detection (BFD).....	258
5.3	Border Gateway Protocol (BGP).....	258
5.4	Broadband Network Gateway (BNG) Control and User Plane Separation (CUPS).....	260
5.5	Certificate management.....	261
5.6	Circuit emulation.....	261
5.7	Ethernet.....	261

5.8	Ethernet VPN (EVPN).....	262
5.9	gRPC Remote Procedure Calls (gRPC).....	263
5.10	Intermediate System to Intermediate System (IS-IS).....	263
5.11	Internet Protocol (IP) Fast Reroute (FRR).....	264
5.12	Internet Protocol (IP) general.....	264
5.13	Internet Protocol (IP) multicast.....	265
5.14	Internet Protocol (IP) version 4.....	267
5.15	Internet Protocol (IP) version 6.....	267
5.16	Internet Protocol Security (IPsec).....	268
5.17	Label Distribution Protocol (LDP).....	270
5.18	Layer Two Tunneling Protocol (L2TP) Network Server (LNS).....	270
5.19	Multiprotocol Label Switching (MPLS).....	271
5.20	Multiprotocol Label Switching - Transport Profile (MPLS-TP).....	271
5.21	Network Address Translation (NAT).....	272
5.22	Network Configuration Protocol (NETCONF).....	272
5.23	Open Shortest Path First (OSPF).....	273
5.24	OpenFlow.....	274
5.25	Path Computation Element Protocol (PCEP).....	274
5.26	Point-to-Point Protocol (PPP).....	274
5.27	Policy management and credit control.....	274
5.28	Pseudowire (PW).....	275
5.29	Quality of Service (QoS).....	275
5.30	Remote Authentication Dial In User Service (RADIUS).....	276
5.31	Resource Reservation Protocol - Traffic Engineering (RSVP-TE).....	276
5.32	Routing Information Protocol (RIP).....	277
5.33	Segment Routing (SR).....	277
5.34	Simple Network Management Protocol (SNMP).....	278
5.35	Timing.....	280
5.36	Two-Way Active Measurement Protocol (TWAMP).....	281
5.37	Virtual Private LAN Service (VPLS).....	281
5.38	Voice and video.....	281
5.39	Wireless Local Area Network (WLAN) gateway.....	282
5.40	Yet Another Next Generation (YANG).....	282
5.41	Yet Another Next Generation (YANG) OpenConfig Modules.....	282

# 1 Getting started

## 1.1 About this guide

This guide describes service mirroring and Operations, Administration and Management (OAM) and diagnostic tools provided by the router and presents examples to configure and implement various tests.

This document is organized into functional chapters and provides concepts and descriptions of the implementation flow, as well as Command Line Interface (CLI) syntax and command usage.



**Note:** Unless otherwise indicated, this guide uses classic CLI command syntax and configuration examples.

The topics and commands described in this document apply to the:

- 7450 ESS
- 7750 SR
- 7950 XRS
- Virtualized Service Router

For a list of unsupported features by platform and chassis, see the *SR OS R23.x.Rx Software Release Notes*, part number 3HE 19269 000 x TQZZA.

Command outputs shown in this guide are examples only; actual displays may differ depending on supported functionality and user configuration.



**Note:**

The SR OS CLI trees and command descriptions can be found in the following guides:

- *7450 ESS, 7750 SR, 7950 XRS, and VSR Classic CLI Command Reference Guide*
- *7450 ESS, 7750 SR, 7950 XRS, and VSR Clear, Monitor, Show, and Tools Command Reference Guide* (for both MD-CLI and Classic CLI)
- *7450 ESS, 7750 SR, 7950 XRS, and VSR MD-CLI Command Reference Guide*



**Note:**

This guide generically covers Release 23.x.Rx content and may contain some content that will be released in later maintenance loads. See the *SR OS R23.x.Rx Software Release Notes*, part number 3HE 19269 000 x TQZZA, for information about features supported in each load of the Release 23.x.Rx software.

## 1.2 Router configuration process

[Table 1: Configuration process](#) lists the tasks necessary to configure mirroring, lawful intercept, and perform tools monitoring functions.

This guide is presented in an overall logical configuration flow. Each section describes a software area and provides CLI syntax and command usage to configure parameters for a functional area.

Table 1: Configuration process

Area	Task	Section
Diagnostics/ Service verification	Mirror implementation	<a href="#">Mirror implementation</a>
	Configure lawful intercept	<a href="#">Lawful Intercept configuration overview</a>
	Configure local and remote end mirror services	<a href="#">Common configuration tasks</a>
	Modify or delete local and remote end mirrored services	<a href="#">Service management tasks</a>
	Troubleshoot services with OAM, SAA, and OAM-PM	<a href="#">OAM fault and performance tools and protocols</a>

## 1.3 Conventions

This section describes the general conventions used in this guide.

### 1.3.1 Precautionary and information messages

The following information symbols are used in the documentation.



**DANGER:** Danger warns that the described activity or situation may result in serious personal injury or death. An electric shock hazard could exist. Before you begin work on this equipment, be aware of hazards involving electrical circuitry, be familiar with networking environments, and implement accident prevention procedures.



**WARNING:** Warning indicates that the described activity or situation may, or will, cause equipment damage, serious performance problems, or loss of data.



**Caution:** Caution indicates that the described activity or situation may reduce your component or system performance.



**Note:** Note provides additional operational information.



**Tip:** Tip provides suggestions for use or best practices.

### 1.3.2 Options or substeps in procedures and sequential workflows

Options in a procedure or a sequential workflow are indicated by a bulleted list. In the following example, at step 1, the user must perform the described action. At step 2, the user must perform one of the listed options to complete the step.

#### **Example: Options in a procedure**

1. User must perform this step.
2. This step offers three options. User must perform one option to complete this step.
  - This is one option.
  - This is another option.
  - This is yet another option.

Substeps in a procedure or a sequential workflow are indicated by letters. In the following example, at step 1, the user must perform the described action. At step 2, the user must perform two substeps (a. and b.) to complete the step.

#### **Example: Substeps in a procedure**

1. User must perform this step.
2. User must perform all substeps to complete this action.
  - a. This is one substep.
  - b. This is another substep.

## 2 Mirror services

When troubleshooting complex operational problems, customer packets can be examined as they traverse the network. Nokia's service mirroring provides the capability to mirror customer packets to allow for trouble shooting and offline analysis. One way to accomplish this is with an overlay of network analyzers established at multiple PoPs, together with skilled technicians to operate them to decode the data provided. This method of traffic mirroring often requires setting up complex filters in multiple switches or routers. These, at best, are only able to mirror from one port to another on the same device.

Nokia's service mirroring extends and integrates these capabilities into the network and provides significant operational benefits. Each router can mirror packets from a specific service to any destination point in the network, regardless of interface type or speed.

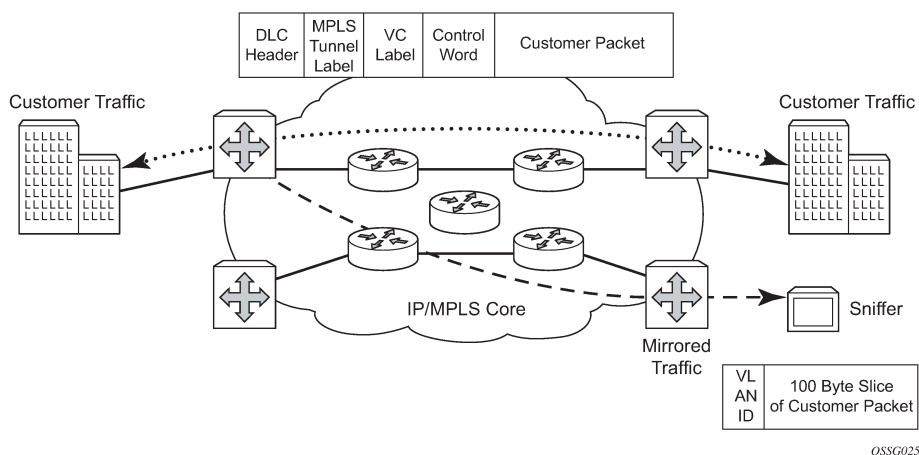
This capability also extends beyond troubleshooting services. Telephone companies can obtain itemized calling records and wire-taps where legally required by investigating authorities. The process can be very complex and costly to carry out on data networks. Service mirroring greatly simplifies these tasks, as well as reduces costs through centralization of analysis tools and skilled technicians.

Nokia routers support service-based mirroring. While some Layer 3 switches and routers can mirror on a per-port basis within the device, Nokia routers can mirror on an n-to-1 unidirectional service basis and re-encapsulate the mirrored data for transport through the core network to another location, using either IP or MPLS tunneling as required ([Figure 1: Service mirroring](#)).

Original packets are forwarded while a copy is sent out the mirrored port to the mirroring (destination) port. Service mirroring allows an operator to see the actual traffic on a customer's service with a sniffer sitting in a central location. In many cases, this reduces the need for a separate, costly overlay sniffer network.

The mirrored frame size that is to be transmitted to the mirror destination can be explicitly configured by using slicing features. This enables mirroring only the parts needed for analysis. For example, only the headers can be copied for analysis, protecting the integrity and security of customer data, or conversely, copying the full packet, including customer data.

*Figure 1: Service mirroring*



## 2.1 Mirror implementation

Mirroring can be implemented on SAPs or ingress network interfaces. The Flexible Fast Path processing complexes preserve the original packet throughout the forwarding and mirroring process, making any necessary packet changes, such as adding encapsulation, on a separate copy.

Mirroring supports multiple types of destinations including local SAPs, remote tunnels, and FTP servers in PCAP format.

Nokia's implementation of packet mirroring is based on the following assumptions:

- Ingress and egress packets are mirrored as they appear on the wire. This is important for troubleshooting encapsulation and protocol issues.
  - When mirroring at ingress, the Flexible Fast Path network processor array (NPA) sends an exact copy of the original ingress packet to the mirror destination while normal forwarding proceeds on the original packet.
  - When mirroring is at egress, the system performs normal packet handling on the egress packet, encapsulating it for the destination interface. A copy of the forwarded packet is forwarded to the mirror destination. Because the mirror copy of the packet is created before egress queuing, the mirrored packet stream may include copies of packets that are discarded in egress queues, such as during congestion or rate limiting.
- Remote destinations are reached by encapsulating the ingress or egress packet within an SDP, like the traffic for distributed VPN connectivity services. At the remote destination, the tunnel encapsulation is removed and the packet is forwarded out a local SAP.

### 2.1.1 Mirror components

Mirroring a packet consists of two components:

- **mirror destinations**

This is where to send the mirrored packet. Various mirror destinations are available and each mirror destination consists of a service ID. Mirrored packets can be sent to a single mirror destination only.

- **mirror sources**

Specify the packets to be mirrored. A mirror source can be configured through **debug** or **service** commands.

### 2.1.2 Mirror source

Mirror sources have the following properties:

- When **config>mirror>mirror-source** and **debug>mirror-source** reference the same mirror source (for example, sap 1/1/1), **config** always takes precedence over **debug**, and **config** removes the **debug** configuration.
- A mirror source can only be mirrored once. It is not possible to send a mirror source to two different mirror destinations.

Ports configured as host ports for satellite and ESA applications are not supported as mirror-source.

Internal ports such as ISA and ESA do not support **config>mirror>mirror-source** and there is only limited support for **debug>mirror**.

### 2.1.2.1 Types and sources

The following commands are supported for **debug** configuration:

- ingress-label
- ip-filter
- ipv6-filter
- isa-aa-group
- mac-filter
- port
- sap
- subscriber

The following commands are supported for **config>mirror>mirror-source**:

- ip-filter
- ipv6-filter
- mac-filter
- port
- sap
- subscriber

Subscriber mirroring applies only to the 7750 SR, 7450 ESS, 7750 SR-s, 7750 SR-e and 7750 SR-a platforms. Enhanced subscriber management associates subscriber hosts with queuing and filtering resources in a shared SAP environment. Mirroring used in subscriber aggregation networks for lawful intercept and debugging is required. Subscriber mirroring capability allows the match criteria to include a subscriber ID.

Subscriber mirroring can also be based on the IP family and host type. The IP family determines if only IPv4 or IPv6 addresses should be mirrored and the host type determines if only IPoE or PPP hosts should be mirrored from the subscriber. To use the IP family and host type, the SAP anti-spoof filter must be set to **ip-mac**. If subscriber mirroring is performed on the L2TP LAC and the IP family is configured as IPv6, no traffic is mirrored for the PPPoE session, even if the LAC subscriber is dual stack. For L2TP LAC, it is recommended that the IP family not be configured or be configured for IPv4 only.

Subscriber mirroring creates a mirror source with subscriber information as match criteria. Specific subscriber packets can be mirrored when using ESM with a shared SAP without prior knowledge of their IP or MAC addresses and without concern that they may change. The subscriber mirroring decision is more specific than a SAP. If a SAP (or port) is placed in a mirror and a subscriber host of which a mirror was configured is mirrored on that SAP, packets matching the subscriber host are mirrored to the subscriber mirror destination.

The mirroring configuration can be limited to specific forwarding classes used by the subscriber. When a forwarding class (FC) map is placed on the mirror, only packets that match the specified FCs are mirrored. A subscriber can be referenced in maximum two different mirror-destinations: one for ingress and one for egress.

Subscriber-based criteria in a mirror source remains in the mirror or LI source configuration even if the subscriber is deleted, removed, or logged off. When the subscriber returns (is configured, created, or logged in) the mirroring resumes. This also implies that a subscriber can be configured as a mirror or LI



source before the actual subscriber exists on the node and before the subscriber ID is active (the mirroring starts when the subscriber is created or logs in and the subscriber ID becomes active).

### 2.1.2.2 Mirror source priority

An operator can configure multiple mirror source services, each asking for the same packets. For instance, using two different mirror source services for a filter and SAPs from the same port. A packet is only mirrored once and in such cases the system selects the highest priority mirror. The mirror source priority, from lowest to highest priority for access ports, is defined below:

1. port mirroring
2. SAP mirroring
3. subscriber mirroring
4. filter mirroring

As an example, when mirroring is enabled on a port for both filter and SAP, packets that matches filter entries rule are mirrored first to the mirror destination for the filter. The rest of the packets that do not match the filter are mirrored to the mirror destination specified for the SAP.

The mirror source priority, from lowest to highest priority for network ports, is defined below:

1. port mirroring
2. label mirroring
3. filter mirroring

### 2.1.3 Mirror destination

Mirror destinations have the following characteristics:

- Mirror destinations can terminate on egress virtual ports which allows multiple mirror destinations to send to the same packet decode device, delimited by IEEE 802.1Q (referred to as Dot1q) tags. This is helpful when troubleshooting a multi-port issue within the network.

When multiple mirror destinations terminate on the same egress port, the individual dot1q tags can provide a DTE/DCE separation between the mirror sources.

- Packets ingressing a port can have a mirror destination separate from packets egressing another or the same port (the ports can be on separate nodes).
- Multiple mirror destinations are supported (local or remote) on a single chassis.
- The operational state of a mirror destination depends on the state of all the outputs of the mirror. The mirror destination goes operationally down if all the outputs are down (for example, all **mirror-dest>sap** and **mirror-dest>spoke-sdp** objects are down, and all gateways configured under **mirror-dest>encap** do not have a known route by which they can be reached). The state of a mirror destination does not depend on inputs such as SDPs configured under **mirror-dest>remote-source**, **debug>mirror-source entries**, or **configure>li>li-source** entries. Some examples of outputs include **mirror-dest>sap** and **mirror-dest>spoke-sdp**.
- **configure>mirror-source** and **li>li-source** can re-use the mirror-destination service that is currently in use by a **debug>mirror-source**. In this scenario, the system automatically cleans up the **debug>mirror-source** entries before it can be re-used. From Release 19.10.R1 onward, config and LI must reference different mirror destinations.

In classic CLI mode, mirror destination supports the following *mirror-type* values:

- ether
- ip-only

In mixed and MD-CLI mode, only the following *mirror-type* values are supported: ether and ip-only.

To switch from classic to mixed or MD-CLI mode, all mirror types other than ether and ip-only must be manually removed first.

The following mirror destinations are supported:

<b>sap</b>	mirroring to a local SAP
<b>spoke-sdp</b>	mirroring to a remote location using a SDP. The remote location uses the remote source to terminate the spoke SDP
<b>remote-source</b>	used at the remote location that is terminating the spoke SDP mirroring tunnel
<b>pcap</b>	mirroring to an FTP server as a PCAP file
<b>encap</b>	mirroring to a remote location as an IP encapsulated packet
<b>endpoint</b>	tunneling redundancy support

### 2.1.3.1 General local and remote mirroring

Mirrored frames can be copied and sent to a specific local destination or service on the router (local mirroring) or copies can be encapsulated and sent to a different router (remote mirroring). This functionality allows network operators to centralize not only network analyzer (sniffer) resources, but also the technical staff who operate them.

The router allows multiple concurrent mirroring sessions so traffic from more than one ingress mirror source can be mirrored to the same or different egress mirror destinations.

Remote mirroring uses an SDP which acts as a logical way of directing traffic from one router to another through a unidirectional service tunnel. The SDP terminates at the far-end router which directs packets to the correct destination on that device.

The SDP configuration from the mirrored device to a far-end router requires a return path SDP from the far-end router back to the mirrored router. Each device must have an SDP defined for every remote router to which it wants to provide mirroring services. SDPs must be created first, before services can be configured.

### 2.1.3.2 Mirror destination type IP-only

The IP mirroring capability for the 7750 SR and 7950 XRS allows a mirror to be created with a parameter that specifies that only the IP packet is mirrored without the original POS/Ethernet DLC header. This results in the mirrored IP packet becoming media-agnostic on the mirror service egress.

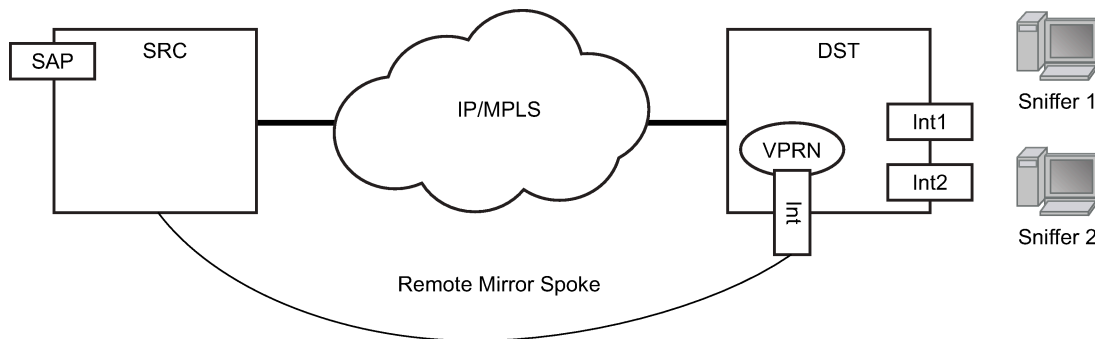
This option is configurable on SAP mirrors for IES, VPRN and VPLS services, Ipipe services, and subscriber mirrors. It is not supported on VLL services such as Epipe, or on ports.

- **remote mirroring termination**

With remote mirroring, the mirror destination configuration can allow IP packets to be mirrored from a source router. The packets are delivered to the destination in a spoke-terminated interface created in a VPRN service. IES interfaces are not supported. The interface can be configured with policy-based routing filters to allow sniffer selection based on incoming mirrored destination IP addresses. The

interface cannot send traffic out as it is a destination-only feature. Packets arriving at the interface are routed based on the routing information within the VPRN. Policy-based routing should always be used unless only a sniffer is connected to the VPRN.

Figure 2: Remote mirroring termination



Fig\_17

- **local mirroring termination**

Local mirroring is like remote mirroring but the source and destination of the mirror exist in the same local IP mirroring node. The configuration must include the source address and destination MAC addresses for the packets going to the sniffer. The destination SAP must be Ethernet.

### 2.1.3.3 Layer 3 encapsulation

The routable encapsulation feature allows mirrored packets to be placed in a routable IP/UDP header and then forwarded in a routing context (either base or VPRN). An additional shim header is also added before the mirrored packet to provide additional context to the collector receiving the packets and contains the direction, mirror type, filter action, interface type, and interface value. This routable encapsulation is available using the **layer-3-encap ip-udp-shim-sampled** command and is supported for ingress and egress mirroring. It can be combined with mirror sampling, slicing, mirror-type ether, and ip-only. [Figure 3: Routable mirror encapsulation](#) shows the routable mirror encapsulation and [Figure 4: Shim header format](#) shows the shim header format.

Figure 3: Routable mirror encapsulation

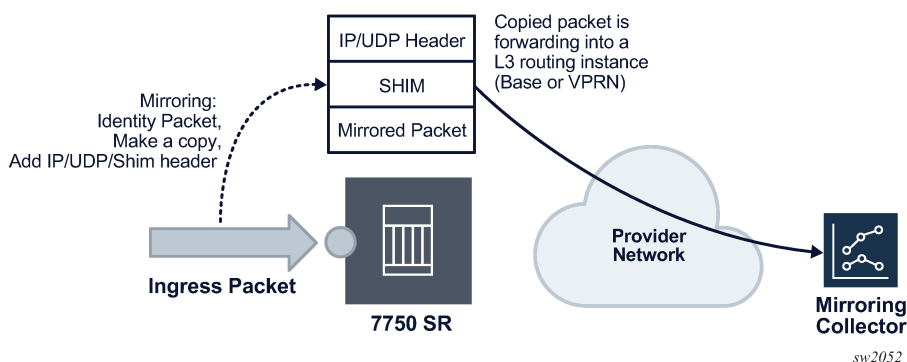
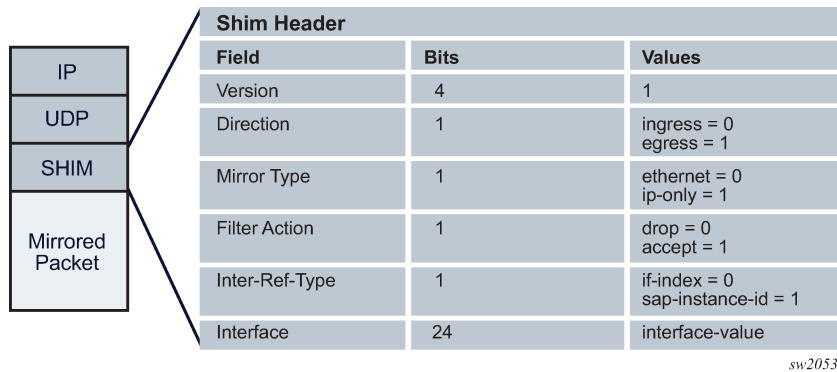


Figure 4: Shim header format



The encoding of the shim header is as follows:

- **Version (4 bits)**  
This describes the shim header version. The only supported version is 1.
- **Direction (1 bit)**  
This describes if the packet was mirrored ingress or egress.
  - Ingress = 0
  - Egress = 1
- **Mirror Type (1 bit)**  
This describes the mirror type.
  - Ethernet = 0
  - IP-Only = 1
- **Filter Action (1 bit)**  
This describes the result of the line card filter action as dropped or accepted. This can be further used by the collector for telemetry purposes.
  - Drop = 0
  - Accept = 1
- **Interface-Ref-Type (1 bit)**  
This indicates whether the interface represents an interface index or a SAP instance.
  - if-index = 0
  - sap-instance-id = 1
- **Interface (24 bits)**  
This can be either a sap-instance-id or the if-index value depending on the interface-ref-type value.

The **if-index** is used in the following mirror source cases:

- For network interface, IES/VRN SAP interface, or interface binding (spoke SDP IP interface), the interface if-index is used.
- For MPLS transit and spoke/mesh-SDP in Layer 2 services, the if-index of the network interface the traffic is received from is used.

- For R-VPLS IP packets to the router interface MAC as well as broadcast and multicast, R-VPLS interface if-index is used.
- For ESM, the subscriber interface if-index is used.

The **sap-instance-id** is used for Layer 2 service SAPs and is an internal reference for the SAP string. The mapping table between the SAP instance ID and SAP strings can be obtained by using the SNMP table `tMirrorSourceSapShimTable`. The `sap-instance-id` found in the shim header can be correlated offline by the collector with `tMirrorSourceSapShimTable` to identify the SAP string and service the packet was mirrored from.

Filter action, interface-ref-type, and interface values are 0 in the case of egress mirroring.

The following restrictions apply to **ip-udp-shim-sampled** encapsulation:

- FP2- and FP3-based cards are not supported as mirror source endpoints. Traffic from these endpoints is not mirrored if they are configured as a mirror source.
- IP UDP shim encapsulation is only supported over IPv4 transport (it is not supported over IPv6 transport).
- Forwarding of routable encapsulated packets from an R-VPLS interface is not supported. Routable encapsulated packets that arrive at the egress of an R-VPLS interface are discarded.
- On the source node where LI mirroring occurs, the operator must configure the mirror destination to inject into the routing instance (that is, base or VPRN) in which the actual destination address is reachable without having to hop into a different instance using GRT leaking. In other words, the interface out of which the packet travels must exist in the routing instance that is configured in the mirror destination.

### 2.1.3.4 PCAP packet capture

#### Prerequisites

Packet capture is a troubleshooting tool that uses both mirroring and debugging. A user's CLI profile must have debug privileges to perform packet capture. To enable packet capture perform the following steps.

#### Procedure

**Step 1.** Set up the mirror destination (in this case, a PCAP). Specify the file URL to which the packet captures are to be sent using the **mirror-dest** command. The packet captures are packaged into the libpcap file format.

The file URL requires the full path, including both username and password, and the filename. When configured, the system performs a syntax check, including an FTP connection test. The configured file URL is rejected if the syntax check fails.

**Step 2.** Specify the source for packet capture. Using either the **debug mirror-source** or **config mirror mirror-source** CLI commands, specify the source to be captured. All mirror sources are supported, including IP-filter, subscriber, SAP, and ports.

The **debug mirror-source** *service-id* must match the **mirror-dest** *service-id* for the PCAP.

**Step 3.** Begin the capture using the **debug pcap session-name capture start** CLI command. The following conditions apply:

- Previous captures with the same filename are overwritten. To avoid a file overwrite, create a new capture with a new filename. This can be accomplished by either renaming the file on the FTP server or by renaming the filename in the mirror destination.

- This CLI command restarts the file transfer session with the remote FTP server.
- If the remote FTP server is unreachable, the command prompt can pause while attempting to re-establish the remote FTP session. The total wait time can be up to 24 seconds (after four attempts of approximately six seconds each).
- If the debug command pauses, verify the following items:
  - the connectivity to the server through the FTP port
  - the FTP user permissions on the FTP server
  - that the FTP server is functional
- The file capture continues indefinitely until the user manually specifies for the packet capture to stop.
- If the file capture fails to start, enter the **show pcap session-name detail** command to see the status of the capture. The detail prompt notifies the operator of the error, and it may require the operator to stop and re start the capture again.

**Step 4.** End the capture. To stop the capture, enter the **debug pcap session-name capture stop** CLI command. This command stops the file transfer session and terminates the FTP session.

- If the FTP server is unreachable, the command prompt rejects further input while it attempts to re-establish the remote FTP session. The total wait time can be up to 24 seconds (4 attempts of approximately 6 seconds each).
- If the **debug** command pauses, check the following items:
  - the connectivity to the server through FTP port
  - the FTP user permissions on the FTP server
  - that the FTP server is functional

## What to do next

The mirrored packets are placed in a buffer in the CPM before they are transferred over FTP or TFTP. The buffer holds a maximum of 20 Mb. The FTP transfer is performed every 0.5 seconds. Each packet that is transferred successfully is flushed from the buffer. Therefore, to ensure all packets are captured successfully, the capture rate must not exceed 20 Mb in 0.5 seconds and the FTP transfer must not exceed 320 Mb/s of bandwidth (20 Mb per 0.5 seconds).

In the following **show pcap** output, the statistics, the session state, write failure, read failures, process time bailouts, and dropped packets are key elements for identifying whether the packet capture on the FTP server is reliable.

```
A:DUT> show pcap "2" detail
=====
Pcap Session "2" Information
=====
Application Type   : mirror-dest      Session State    : ready
Capture           : stop                Last Changed    : 02/06/2018 19:52:07
Capture File Url  : ftp://*:*@192.168.41.1/pcap2.pcap
Buffer Size      : 10 Bytes         File Size       : 200 Bytes
Write Failures   : 0                Read Failures   : 0
Proc Time Bailouts : 0                Last File Write : 02/06/2018 19:52:07
Dropped Packets  : 661 Packets
=====
```

Packet capture is a troubleshooting tool. Therefore, all CLI commands except for the FTP URL destination are located under **debug**. This allows the administrator to set up a CLI profile specifically for packet capture with debug privileges.

The packet capture uses FTP for file transfer and can be routed to the destination using the management port or through the IOM port. If the FTP server destination is routed through the management port, consider the maximum bandwidth available.



**Caution:** Typically, the management port is used for logging, SNMP, SSH/Telnet, AAA, and other management services. A high-throughput packet capture may disrupt these management services. Therefore, use packet capture transfers using the management port with caution.

Mechanisms are built in to prevent mirroring or packet captures that result in loops or daisy-chains. However, it is possible to form a loop or daisy-chain if routing re routes or configuration changes. When a packet capture becomes looped or daisy-chained, the packet capture stops.



**Note:** When executing an **admin rollback** for a configuration under the **config mirror mirror-dest service-id pcap** CLI context, the **pcap** must first be stopped by executing the **debug pcap session-name capture stop** command. If the **pcap** is not stopped, the **rollback** fails.

### 2.1.3.5 Mirrored traffic transport using MPLS-TP SDPs

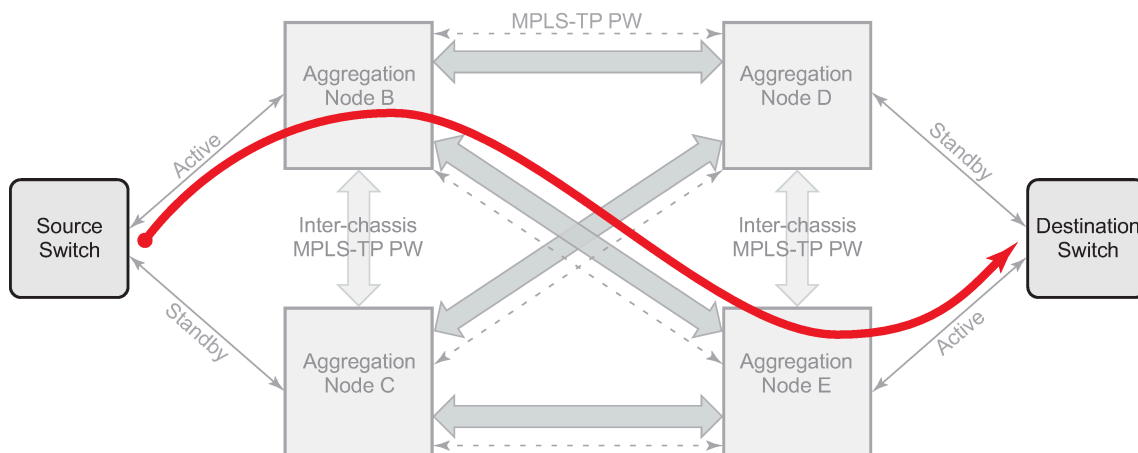
Bidirectional MPLS-TP spoke SDPs with a configured pw-path-id can transport a mirrored service. Mirror services are not supported on static PWs with an MPLS-TP pw-path-id bound to an SDP that uses an RSVP-TE LSP.

Mirror services using MPLS-TP spoke SDPs can be configured using CLI in the context `mirror-dest>remote-source`. For both the CPM and IOM, this enables reuse of spokes for mirror services and other services such as pipes.

Control channel status signaling is supported with PW redundancy on spoke SDPs in a mirror context.

The following is an example of PW redundancy for a mirror service. In this case, MPLS-TP spoke SDPs are used.

Figure 5: Mirroring with PW redundancy using MPLS-TP



al\_0526

Note that mirroring traffic is usually unidirectional, flowing from "source" nodes (B or C) to "destination" nodes (D or E). However, in case of MPLS-TP, the control channel status packets may flow in the reverse direction.

The following is an example of a mirror service configuration using MPLS-TP spoke SDPs:

### Source node B

```
#-----
echo "Mirror Configuration"
#-----
mirror
  mirror-dest 300 create
  endpoint "X" create
  revert-time 100
  exit
  endpoint "Y" create
  revert-time 100
  exit
  remote-source
  spoke-sdp 230:1300 endpoint "Y" icb create
  ingress
  vc-label 13301
  exit
  egress
  vc-label 13301
  exit
  control-word
  pw-path-id
  agi 1:1
  saii-type2 1:10.20.1.2:13301
  taii-type2 1:10.20.1.3:13301
  exit
  control-channel-status
  refresh-timer 10
  no shutdown
  exit
  no shutdown
  exit
  exit
  spoke-sdp 240:300 endpoint "X" create
  ingress
  vc-label 2401
  exit
  egress
  vc-label 2401
  exit
  control-word
  pw-path-id
  agi 1:1
  saii-type2 1:10.20.1.2:2401
  taii-type2 1:10.20.1.4:2401
  exit
  control-channel-status
  refresh-timer 10
  no shutdown
  exit
  no shutdown
  exit
  spoke-sdp 250:300 endpoint "X" create
  ingress
  vc-label 6501
  exit
  egress
```



```

        vc-label 6501
    exit
    control-word
    pw-path-id
    agi 1:1
    sai-type2 1:10.20.1.2:6501
    taii-type2 1:10.20.1.5:6501
    exit
    control-channel-status
    refresh-timer 10
    no shutdown
    exit
    no shutdown
exit
spoke-sdp 230:300 endpoint "X" icb create
    ingress
        vc-label 12301
    exit
    egress
        vc-label 12301
    exit
    control-word
    pw-path-id
    agi 1:1
    sai-type2 1:10.20.1.2:12301
    taii-type2 1:10.20.1.3:12301
    exit
    control-channel-status
    refresh-timer 10
    no shutdown
    exit
    no shutdown
    exit
    no shutdown
    exit
    no shutdown
exit
exit
exit all

```

### Destination node C

```

#-----
echo "Mirror Configuration"
#-----
    mirror
        mirror-dest 300 create
        endpoint "X" create
        revert-time 100
    exit
    endpoint "Y" create
        revert-time 100
    exit
    remote-source
        spoke-sdp 230:1300 endpoint "Y" icb create
        ingress
            vc-label 13301
        exit
        egress
            vc-label 13301
        exit
        control-word
        pw-path-id
        agi 1:1
        sai-type2 1:10.20.1.3:13301

```

```
        taii-type2 1:10.20.1.2:13301
    exit
    control-channel-status
        refresh-timer 10
        no shutdown
    exit
    no shutdown
exit
exit
spoke-sdp 340:300 endpoint "X" create
    ingress
        vc-label 6501
    exit
    egress
        vc-label 6501
    exit
    control-word
    pw-path-id
        agi 1:1
        saii-type2 1:10.20.1.3:6501
        taii-type2 1:10.20.1.4:6501
    exit
    control-channel-status
        refresh-timer 10
        no shutdown
    exit
    no shutdown
exit
spoke-sdp 350:300 endpoint "X" create
    ingress
        vc-label 2401
    exit
    egress
        vc-label 2401
    exit
    control-word
    pw-path-id
        agi 1:1
        saii-type2 1:10.20.1.3:2401
        taii-type2 1:10.20.1.5:2401
    exit
    control-channel-status
        refresh-timer 10
        no shutdown
    exit
    no shutdown
exit
spoke-sdp 230:300 endpoint "X" icb create
    ingress
        vc-label 12301
    exit
    egress
        vc-label 12301
    exit
    control-word
    pw-path-id
        agi 1:1
        saii-type2 1:10.20.1.3:12301
        taii-type2 1:10.20.1.2:12301
    exit
    control-channel-status
        refresh-timer 10
        no shutdown
    exit
```

```

        no shutdown
    exit
    no shutdown
exit
exit

```

### Source node D

```

#-----
echo "Mirror Configuration"
#-----
mirror
  mirror-dest 300 create
  endpoint "X" create
  revert-time 100
  exit
  endpoint "Y" create
  revert-time 100
  exit
  remote-source
    spoke-sdp 240:300 endpoint "Y" create
    ingress
      vc-label 2401
    exit
    egress
      vc-label 2401
    exit
    control-word
    pw-path-id
      agi 1:1
      saii-type2 1:10.20.1.4:2401
      taii-type2 1:10.20.1.2:2401
    exit
    control-channel-status
      refresh-timer 10
      no shutdown
    exit
    no shutdown
  exit
  spoke-sdp 340:300 endpoint "Y" create
  ingress
    vc-label 6501
  exit
  egress
    vc-label 6501
  exit
  control-word
  pw-path-id
    agi 1:1
    saii-type2 1:10.20.1.4:6501
    taii-type2 1:10.20.1.3:6501
  exit
  control-channel-status
    refresh-timer 10
    no shutdown
  exit
  no shutdown
exit
  spoke-sdp 450:1300 endpoint "Y" icb create
  ingress
    vc-label 13301
  exit
  egress

```

```

        vc-label 13301
    exit
    control-word
    pw-path-id
    agi 1:1
    saii-type2 1:10.20.1.4:13301
    taii-type2 1:10.20.1.5:13301
    exit
    control-channel-status
    refresh-timer 10
    no shutdown
    exit
    no shutdown
    exit
    sap lag-10:300.1 endpoint "X" create
    exit
    spoke-sdp 450:300 endpoint "X" icb create
    ingress
    vc-label 12301
    exit
    egress
    vc-label 12301
    exit
    control-word
    pw-path-id
    agi 1:1
    saii-type2 1:10.20.1.4:12301
    taii-type2 1:10.20.1.5:12301
    exit
    control-channel-status
    refresh-timer 10
    no shutdown
    exit
    no shutdown
    exit
    no shutdown
    exit
    exit

```

### Destination node E

```

#-----
echo "Mirror Configuration"
#-----
    mirror
        mirror-dest 300 create
            endpoint "X" create
            revert-time 100
        exit
        endpoint "Y" create
            revert-time 100
        exit
        remote-source
            spoke-sdp 250:300 endpoint "Y" create
            ingress
            vc-label 6501
        exit
        egress
            vc-label 6501
        exit
        control-word
        pw-path-id

```

```
        agi 1:1
        saii-type2 1:10.20.1.5:6501
        taii-type2 1:10.20.1.2:6501
    exit
    control-channel-status
        refresh-timer 10
        no shutdown
    exit
    no shutdown
exit
spoke-sdp 350:300 endpoint "Y" create
    ingress
        vc-label 2401
    exit
    egress
        vc-label 2401
    exit
    control-word
    pw-path-id
        agi 1:1
        saii-type2 1:10.20.1.5:2401
        taii-type2 1:10.20.1.3:2401
    exit
    control-channel-status
        refresh-timer 10
        no shutdown
    exit
    no shutdown
exit
spoke-sdp 450:1300 endpoint "Y" icb create
    ingress
        vc-label 13301
    exit
    egress
        vc-label 13301
    exit
    control-word
    pw-path-id
        agi 1:1
        saii-type2 1:10.20.1.5:13301
        taii-type2 1:10.20.1.4:13301
    exit
    control-channel-status
        refresh-timer 10
        no shutdown
    exit
    no shutdown
exit
exit
sap lag-10:300.1 endpoint "X" create
exit
spoke-sdp 450:300 endpoint "X" icb create
    ingress
        vc-label 12301
    exit
    egress
        vc-label 12301
    exit
    control-word
    pw-path-id
        agi 1:1
        saii-type2 1:10.20.1.5:12301
        taii-type2 1:10.20.1.4:12301
    exit
```

```
control-channel-status
  refresh-timer 10
  no shutdown
exit
no shutdown
exit
no shutdown
exit
exit
```

### 2.1.3.6 Slicing and sampling

Slicing and sampling are available to all mirror destinations:

- **slicing**

Slicing copies a specified packet size of each frame. This is useful to monitor network usage without having to copy the actual data. Slicing enables mirroring larger frames than the destination packet decode equipment can handle. It also allows conservation of mirroring resources by limiting the size of the stream of packet through the router and the core network.

When a mirror slice size is defined, a threshold that truncates a mirrored frame to a specific size is created. For example, if a value of 256 bytes is defined, up to the first 256 bytes of the frame are transmitted to the mirror destination. The original frame is not affected by the truncation. Mirrored frames, most likely, become larger as encapsulations are added when packets are transmitted through the network core or out the mirror destination SAP to the packet or protocol decode equipment. Slice size is not supported by CEM encap-types or IP mirroring.

The transmission of a sliced or non-sliced frame is also dependent on the mirror destination SDP path MTU and the mirror destination SAP physical MTU. Packets that require a larger MTU than the mirroring destination supports are discarded if the defined slice size does not truncate the packet to an acceptable size.

- **sampling**

Mirror sampling rate defines a packet sampling rate for a mirror service. The sampling rate is applicable to all endpoints in the mirror source ingress and egress and supported on FP4 and FP5-based cards.

This capability can be useful for analytics purposes such as DDoS telemetry to provide a subset of traffic while still maintaining statistical accuracy using packet sampling.

Packet sampling can be configured concurrently with mirror slicing to further limit the amount of traffic sent to the collector.

For endpoints in the mirror source on FP2- and FP3-based cards, all the packets are mirrored without sampling.

#### 2.1.3.6.1 Global sampling rate

SR OS supports three types of sampling rates:

- **High-rate sampling**

High-rate sampling allows the sampling of 1 packet out of every 2 to 255 packets. Use the following command to configure high-rate sampling.

```
configure mirror global-sampling-rate
```

Optionally, each mirror destination service can adopt the global sampling rate, allowing one single high-rate sampling rate for the entire system, by using the following command.

```
configure mirror mirror-dest use-global-sampling-rate
```

- **Low-rate sampling**

Low-rate sampling allows the sampling of 1 packet out of every 256 to 10,000 packets. Unlike high-rate sampling, each mirror destination can use a different low-sampling rate. Use the following command to configure low-rate sampling for each destination.

```
configure mirror mirror-dest sampling-rate
```

- **No sampling**

When neither the **use-global-sampling-rate** or **sampling-rate** commands under the mirror destination, the system mirrors all packets to the destination.



**Note:**

Each mirror destination can use a different low-sampling rate. However, if the high-sampling rate is configured, using the **global-sampling-rate** command, all mirror destinations share the same high-sampling rate.

When both the low-rate and high-rate are configured under the same mirror destination, the low rate takes precedence. The system automatically samples using the low rate specified and ignores the global high rate.

## 2.1.4 Mirroring performance

Replication of mirrored packets can, typically, affect performance and should be used carefully. Nokia routers minimize the impact of mirroring on performance by taking advantage of its distributed Flexible Fast Path technology. Flexible Fast Path forwarding allows efficient mirror service scaling and, at the same time, allows a large amount of data to be mirrored with minimal performance impact. When a mirror destination is configured, the packet slice option can truncate mirrored packets to the destination, which minimizes replication and tunneling overhead.

## 2.2 Lawful Intercept

Lawful Intercept (LI) describes a process to intercept telecommunications by which law enforcement authorities can unobtrusively monitor voice and data communications to combat crime and terrorism with higher security standards of lawful intercept capabilities in accordance with local law and after following due process and receiving authorization from competent authorities. The interception capabilities are sought by various telecommunications providers.

As lawful interception is subject to national regulation, requirements vary from one country to another. Nokia's implementation satisfies most national standard's requirements. LI capability is configurable for all Nokia service types.

LI mirroring is configured by an operator that has LI permission. LI mirroring is hidden from anyone who does not have the right permission.

In Release 19.10.R1 and higher config and LI must use different mirror destinations.

## 2.2.1 LI activation through RADIUS

In addition to CLI and SNMP control, RADIUS messages also activate LI sessions for subscriber-host targets. Activation through RADIUS is equivalent to adding or removing a set of subscriber-host entries in an LI source.



**Note:** The term "activation" in this section represents both "activation and de-activation".

The activation of an LI session via RADIUS applies to the 7450 ESS and 7750 SR and can occur in one of two ways:

- when the RADIUS Access-Accept message is received by the 7450 ESS or 7750 SR. The target (either a host or a set of hosts) is implicit. The target acts as the same host (or set of hosts) that is within the scope of the Access-Accept and interception occurs for this entire set of hosts (or a single host).
- through RADIUS CoA messages. The target (set of hosts) is identified through one of the following methods:
  - Acct-Session-Id (which can represent a single host or a collection of hosts)
  - a *sap-id;ip-addr* carried in the NAS-Port-Id (attr 87) and the Framed-Ip-Address (attr 8)." for IPv4 hosts
  - a *sap-id;IPv6\_addr* carried in the NAS-Port-ID (attr 87) and one of Alc-Ipv6-Address, Framed-Ipv6-Prefix, or Delegated-Ipv6-Prefix for IPv6 hosts
  - Alc-Subsc-ID-Str

The following set of VSAs is used to activate LI sessions via RADIUS:

- Alc-LI-Action – ON/OFF/NONE
- Alc-LI-Destination - <string> and has two options:
  - the mirror destination service ID
  - at real time, specify the IP destination, the UDP port, and the router instance of the LI mediation device

The format for the VSA is **ip-address** [:port] [**router instance**]. The IP address must be of type IPv4 and is the only mandatory parameter.

- Alc-LI-Direction – INGRESS/EGRESS
- Alc-LI-FC – be/l1/l2/af/ef
- (optional) Alc-LI-Use-Outside-IP

Use this VSA when the subscriber is an L2-aware NAT subscriber and uses the outside IP address instead of the private IP address for packet mirroring. See [L2-Aware NAT](#) for more details.

The Alc-LI-FC VSA can be present several times if more than one forwarding class (FC) is subject to LI.



The VSAs Alc-LI-Direction and Alc-LI-FC are optional. If either is not included, both directions (ingress and egress) as well as all FCs are mirrored.

The Alc-LI-Destination VSA can be used in one of the following ways:

- A mirror destination must first be provisioned on SR. To use the mirror destination, the VSA specifies the mirror destination service ID in the Access-Accept message or a CoA.
- The VSA specifies the IP address of the mirror destination through the Access-Accept message or a CoA. The reserved range of service IDs and the mirror destination template must be configured first. This VSA provisions the mirror destination using a combination of parameters from the LI template and RADIUS VSAs. The following should be considered when using this VSA:
  - Only Layer 3 encapsulation is supported as the mirror destination.
  - The VSA has the format *ipv4-address* [:*port*] [**router** {**Base** | *svc-id*}]. The VSA must include the LI destination IPv4 address, while the port and the routing instance are optional. If the destination port and routing instance are not specified in the VSA, the configuration from the LI mirror destination template is used.
  - With the LI mirror destination reservation, a list of service IDs is reserved for configuring the mirror destination via RADIUS. The LI mirror destination is shared with the mirror destination used for debugging purposes. Therefore, it is suggested to reserve enough for LI purposes, and leave enough for debugging and configuration. The VSA triggers the creation of a mirror destination automatically and uses one of the service IDs in the reservation range. An LI source that matches the IP source, IP destination, UDP destination, UDP source, and direction bit, reuses the same LI mirror destination service ID. The LI mirror destination reservation range can be expanded or reduced in real time. The range can be changed completely when there are no LI sources referenced in the mirror reservation range.
  - The LI mirror destination template specifies the parameters for the Layer 3 encapsulation. It is mandatory to provision the IP source, IP destination, UDP source, and UDP destination parameters.
  - It is possible to configure up to eight LI mirror destination templates. The mirror destination template can be switched in real time, if, for example, a parameter such as the source IP address is to be updated.
  - The system can block RADIUS from generating the mirror destination by removing a template reference under the **config>li>radius** context.

VSAs in the Access-Accept messages also activate LI for a newly-created host. In this case, the LI activation is not addressed by the Acct-Session-Id, as this is not yet known during session authorization.

Different attributes can be used in a CoA to identify one or more subscriber hosts. Typically, only a single attribute or set of attributes is used to target a host or several: NAS-Port-Id + IP, Acct-Session-Id, or Alc-Subsc-ID-Str. In the case where "NAS-Port-Id + IP" is used in a Wholesale or Retail model, the Alc-Retail-Serv-Id VSA must be included in the CoA.

The ability to delete all **li-source** entries from a mirror service is also available via RADIUS. This function may be useful when an LI mediation device loses synchronization with the SR OS state and needs to reset a mirror service to a known state with no LI sessions. This clear function is performed by sending the following attributes in a RADIUS CoA. If the CoA does not contain exactly the following three VSAs (each with a valid value matching the configuration on SR OS), the CoA is silently dropped without a NAK:

- **Alc-LI-Action**  
Alc-LI-Action = 'clear-dest-service'
- **Alc-LI-Destination**

The destination can specify the service ID of the mirror destination or it can pass the VSA in the mirror destination IP, where the mirror destination IP was automatically created by RADIUS.

- Alc-LI-Destination = *service-id*, if a mirror destination service ID was used for LI
- Alc-LI-Destination = **ip-address** [*:port*] [*router instance*]. The system deletes RADIUS auto-generated mirror destinations based on three parameters: the IP destination, the UDP destination port, and the router instance. These parameters can be passed in from the Alc-LI-Destination VSA. If the VSA provides only some of the parameters, for example, only the destination IP, the parameters from the mirror destination template is used (from **config>li>mirror-dest-template**). The three parameters determine the mirror service ID to delete and any combination of the IP source, UDP source port, and direction bit can be deleted. It is possible that a template change can prevent the VSA from deleting the mirror destination service. To manually delete a mirror destination, a CLI command is provided under **clear li radius mirror-dest svc-id**. To determine the service ID to delete, a manual login is required.

- **Alc-Authentication-Policy-Name**

This VSA is only required in a specific configuration. The VSA is not required when a RADIUS server policy is configured under **configure subscriber-mgmt authentication-policy** and the RADIUS server policy servers are used as CoA servers.

This VSA is required in the configuration where the servers configured inside the authentication policy are used as CoA servers, with the following:

- a list of servers is configured under **config>subscr-mgmt>auth-plcy>radius-auth-server**
- **accept-authorization-change** is enabled under **config>subscr-mgmt>auth-plcy**
- the authentication policy does not reference the RADIUS server policy

When the above conditions are met, the Alc-Authentication-Policy-Name VSA is required and must reference the authentication policy that contains the IP address of the LI CoA client.

The LI-related VSAs cannot be combined in one CoA message with other action-related VSAs (force renew, change of SLA profile, and so on). The only exception to this rule is for the CoA used to create a new subscriber host. In this case, LI-related VSAs can be included, along with other VSAs.

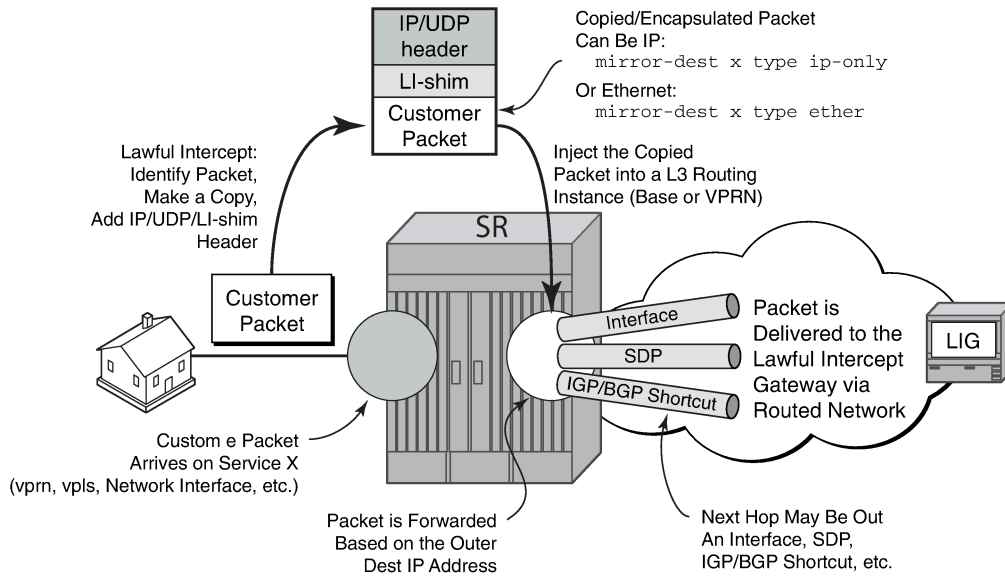
If LI is activated through CLI or SNMP, the activation through RADIUS takes precedence. The precedence in this context means that RADIUS activation of LI fully overrides whatever was configured at CLI or SNMP level for this host. If the RADIUS LI is de-activated, the CLI or SNMP configuration becomes active again.

The LI-related VSAs are not shown in debug messages. The **show li li-source** command shows all sub-hosts for which LI was activated using RADIUS VSAs. This command is only accessible to CLI users with LI privileges.

## 2.2.2 Routable Lawful Intercept encapsulation

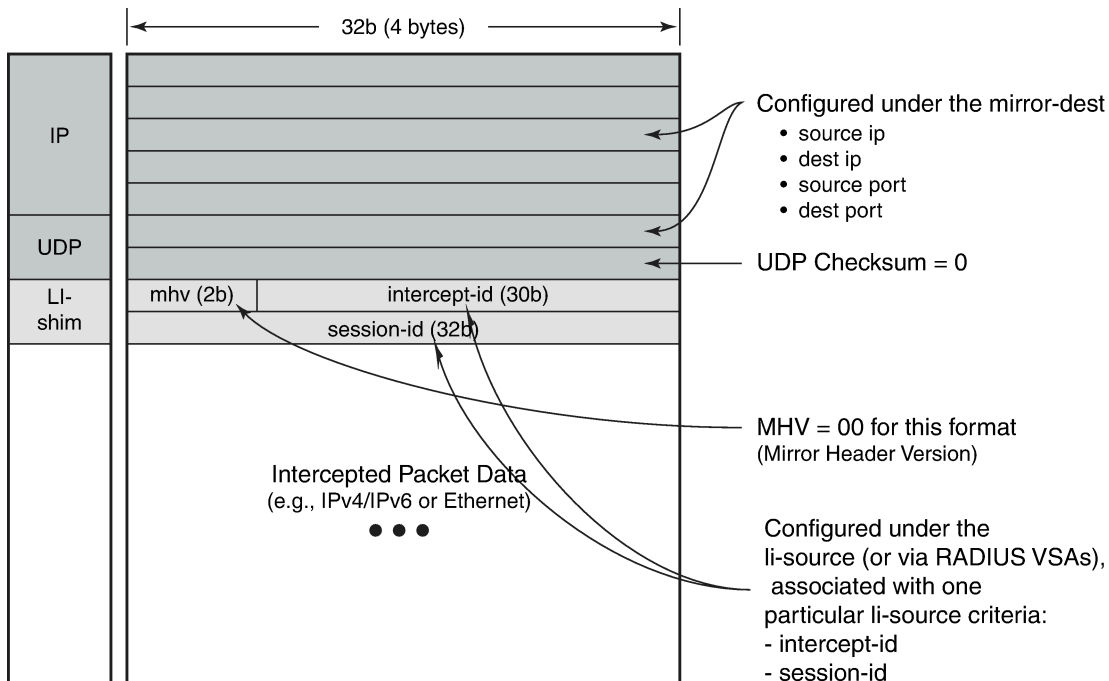
The Routable LI encapsulation feature allows LI mirrored packets to be placed into a routable (for example, IP/UDP) header and then forwarded in a routing context (base or VPRN). An LI-shim inserted before the customer packet allows correlation of packets to LI sessions at the downstream LI Mediation device (LIG).

Figure 6: Routable Lawful Intercept encapsulation



OSSG687

Figure 7: Routable encapsulation format

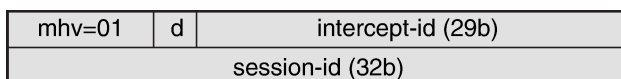


OSSG685

Some of the supported attributes and scenarios for the routable LI encapsulation feature include the following:

- The part of the customer packet that is copied and placed into the routable encapsulation can be either the IP packet (with none of the original Layer2 encap) or an Ethernet packet by selecting either **ip-only** or **ether** as the mirror-dest type.
- The ability to inject into the Base routing instance (for forwarding out network interfaces or IES SAPs for example) or a VPRN service.
- The ability to forward the encapsulated packets out VPRN SDPs, IGP/BGP shortcuts and SDP spoke interfaces.
- Options to use ip, udp, li-shim or ip, gre routable encapsulation (applies to the 7450 ESS and 7750 SR).
- An optional direction bit in the li-shim; if the use of the direction bit is configured, then a bit from the **intercept-id** (configure under **mirror-dest**) is "stolen". Only a 29b **intercept-id** is allowed for **li-source** entries if the mirror destination is configured to use a direction bit.

Figure 8: LI-shim version 01 with a direction bit



OSSG686

- The encoding of the direction (d) bit is as follows:
  - 0 = ingress
  - 1 = egress
- For NAT based LI, ingress means the traffic arriving at the node from the subscriber host (applies to the 7450 ESS and 7750 SR).
- User configurable **intercept-id** and **session-id** per li-source entry that is placed into the li-shim (a total max of 62 configurable bits).
- Configuration via CLI/SNMP or RADIUS (applies to the 7450 ESS and 7750 SR). For RADIUS configuration the following VSAs are used:
  - Alc-LI-Action, Alc-LI-Direction, Alc-LI-Destination, Alc-LI-FC (See [LI activation through RADIUS](#)).
  - Alc-LI-Intercept-Id specifies the intercept-id to place in the LI shim. Only applicable if the mirror-dest (as specified by the Alc-LI-Destination) is configured with routable encap that contains the LI-Shim. A value of 0 is used if this VSA is not present.
  - Alc-LI-Session-Id specifies the session-id to place in the LI-Shim. Only applicable if the mirror-dest (as specified by the Alc-LI-Destination) is configured with routable encap that contains the LI shim. A value of 0 is used if this VSA is not present.
- A LI session configured via RADIUS takes precedence over a session configured via CLI, but the CLI mirror is re-instated if the RADIUS mirror request is later removed (applies to the 7450 ESS and 7750 SR)
- ip, udp, and li-shim encap is available for ether and LI shim mirror-dest types.
- ip | udp | li-shim encap is available for all li-source entry types (sap, filter, subscriber and nat).



**Note:** For NAT based Lawful Intercept, routable LI encap is available, as well as the MAC or Layer 2-based encapsulation for NAT LI as configured under **config>li>li-source>nat>ethernet-encap** (applies to the 7450 ESS and 7750 SR)

- Fragmentation of the resulting mirror packet is supported. Note that fragmentation is supported for NAT LI with the routable encapsulation, but fragmentation is not supported for NAT LI with Ethernet encapsulation (applies to the 7450 ESS and 7750 SR).

The following restrictions apply to the routable LI encapsulation feature:

- Only applicable to Lawful Intercept and is not available for debug or MS-ISA based Application Assurance mirrors. MS-ISA based Application Assurance is applicable to the 7450 ESS and 7750 SR.
- IPv4 transport only (the routable encapsulation cannot be IPv6).
- On the mirror source node, forwarding of routable encapsulated LI packets out of an R-VPLS interface is not supported. A mirror destination configured with routable encapsulation can be bound to a routing instance that also has an R-VPLS bound to it, but the operator must ensure that the destination of the LI packets is not reachable via any R-VPLS interfaces. Any routable encapsulated LI packets that arrive at the egress of an R-VPLS interface are discarded. Parallel use of routable LI encapsulation and R-VPLS in the same routing instance is supported if the mirrored packets do not egress out of the R-VPLS interface.
- `ip | gre encap` is supported for the **ip-only** mirror destination type only, and only for subscriber li-source entries (CLI, SNMP, or RADIUS based). Subscriber management is not supported on the 7950 XRS.

The contents of the GRE header are all zeros (all optional bits zero, no optional headers/fields like checksum, offset, key, seq, and so on) except for the Protocol field which contains 0x0800 for IPv4 packets or 0x86DD for IPv6 packets. The far-end receiver of the intercepted packets must be configured to expect no GRE options (that is, no key, no checksum, and so on).

- On the source node where LI mirroring occurs, the operator must configure the mirror-dest to inject into the routing instance (that is, base or VPRN) in which the actual destination address is reachable without having to hop into a different instance using GRT leaking. In other words, the interface out, which the packet travels, must exist in the routing instance that is configured in the mirror-dest.

For example, if the LIG is at 110.120.130.140 and is in the base instance, but VPRN-1 has a default route to the GRT (for example, 0.0.0.0->GRT) then the operator must configure the mirror destination to inject into the base (even though theoretically address 110.120.130.140 is reachable from VPRN-1). If the operator attempts to configure the mirror destination to inject into VPRN-1, and VPRN-1 itself does not have reachability to 110.120.130.140 out an interface that is part of the VPRN, then the mirror destination is operationally down.

Care must be taken in the configuration of LI mirrors and the destination IP address for the routable LI encapsulation. Incorrect selection of the destination IP could send packets to unintended destinations (for example, configuring the encapsulation with a subscriber's IP address), and combinations of mirrors and routable encapsulation can create loops in the network.

## 2.2.3 Lawful Intercept and NAT

### 2.2.3.1 Carrier grade NAT

LI for NAT is supported to mirror configured subscriber's traffic to a mirror destination. When active, packets are mirrored from the perspective of the NAT outside interface (after NAT translations have occurred). All traffic for the specified subscriber, including traffic associated with static port-forwards, is mirrored. This feature is supported for 7450 ESS and 7750 SR only.

A simplified Ethernet encapsulation (with an optional Intercept ID) is used for all NAT traffic. When mirroring NAT traffic, the mirror destination must be of type **ether**. The customer packet from the (outside)

IP header onwards (including the IP header) is mirrored. The operator has the configuration option of embedding the intercept ID into the LI packet using an explicit **intercept-id** command. Both packet formats are described below:

Figure 9: Ethernet mirror examples

Standard Ethernet Mirror:

Ethernet	Destination MAC Address...	
	...Destination MAC Address	Source MAC Address...
	...Source MAC Address	
H	Ethertype (IPv4 = 0x0800)	... customer packet. i.e. IPv4

Ethernet Mirror with optional Intercept ID:

Ethernet	Destination MAC Address...	
	...Destination MAC Address	Source MAC Address...
	...Source MAC Address	
LI	Ethertype (configurable)	Intercept ID...
	...Intercept ID	Ethertype (IPv4 = 0x0800)
H	... customer packet. i.e. IPv4	

OSSG539

The contents of the highlighted fields are configurable using the following CLI:

```
li
  li-source service-id
  nat
    classic-lsn-sub router name ip address
      intercept-id id
    dslite-lsn-sub router name b4 ipv6-address
      intercept-id id
    l2-aware-sub sub-ident
      intercept-id id
```

The default Ethernet-header is to use etype 0x600 and system MAC address for both the source and destination addresses. The configurable Ethertype and Intercept ID is only added when an intercept ID is present for the subscriber in the NAT configuration.

### 2.2.3.2 L2-Aware NAT

When Layer 3 encapsulation is configured as the mirror destination for an L2-Aware NAT subscriber, the mirror destination must be of type **ip-only** and the encapsulation must be of type **ip-udp-shim**. For L2-Aware NAT, it is possible to assign the same inside IPv4 private IP address to all subscribers. It is preferable to intercept the L2-Aware NAT subscriber using the outside IP address instead. This can be accomplished from both RADIUS and CLI as described in the following table.

Table 2: Use of inside and outside IPs for LI

	Lawful Intercept to use host inside IP address	Lawful Intercept to use host outside IP address
CLI access	<ol style="list-style-type: none"> <li>1. Configure the subscriber ID under <b>config&gt;li&gt;li-source&gt;nat&gt;l2-aware-sub.</b></li> <li>2. Configure the LI IP filter through the subscriber SLA profile.</li> </ol> <p>The command <b>config&gt;li&gt;use-outside-ip-address</b> does not apply to CLI configured LI targets.</p>	<p>Configure the subscriber ID under <b>config&gt;li&gt;li-source&gt;nat&gt;l2-aware-sub.</b></p> <p>The command <b>config&gt;li&gt;use-outside-ip-address</b> does not apply to CLI configured LI targets.</p>
RADIUS access	<ol style="list-style-type: none"> <li>1. Ensure <b>config&gt;li&gt;use-outside-ip-address</b> is disabled. Use RADIUS Acct-Session-Id, subscriber-id, and so on, to enable the LI session.</li> <li>2. If <b>config&gt;li&gt;use-outside-ip-address</b> is enabled, when enabling LI via RADIUS, the VSA "Alc-LI-Use-Outside-IP = false" must be included.</li> </ol>	<ol style="list-style-type: none"> <li>1. Ensure <b>config&gt;li&gt;use-outside-ip-address</b> is enabled. Use RADIUS Acct-Session-Id, subscriber-id, and so on, to enable the LI session.</li> <li>2. If <b>config&gt;li&gt;use-outside-ip-address</b> is disabled, when enabling LI via RADIUS, the VSA "Alc-LI-Use-Outside-IP = true" must be included.</li> </ol>

When the RADIUS VSA Alc-LI-Use-Outside-IP is used, the configuration **config>li>use-outside-ip-address** is ignored.

Alc-Use-Outside-IP is only supported when the mirror destination service is configured with Layer 3 encapsulation.

L2-Aware subscribers do not support the LI RADIUS VSAs Alc-LI-FC and Alc-LI-Direction. When an L2-Aware subscriber is subjected to LI via CLI or RADIUS, dual stack traffic is mirrored.

## 2.2.4 Lawful Intercept management interfaces

LI can be managed using classic management interfaces (for example, classic CLI or SNMP) or model-driven management interfaces (MD-CLI or NETCONF). Management of LI is similar across all interfaces.

See "Classic and Model-Driven Management Interfaces" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR System Management Guide* for more information about management interfaces and setting the configuration mode.

### 2.2.4.1 LI management using the classic CLI engine

With the advent of support for both classic and mixed configuration modes for LI management, the classic engine supports the following additional features in classic configuration mode.

Classic CLI mode features for LI management are as follows:

- LI filter names, as well as filter IDs, under **li-filter-associations**, including **ip-filter-name**, **ipv6-filter-name**, and **mac-filter-name**
  - Operators can choose between IDs and names (mutually exclusive).
  - IDs must be hard references and can only refer to filters (IP, IPv6, MAC) that already exist.
  - Names can be loose references and can refer to filters (IP, IPv6, MAC) that do not exist.
- LI filter names, as well as filter IDs, under **li-filter-block-reservation**, including **ip-filter-name**, **ipv6-filter-name**, and **mac-filter-name**
  - Operators can choose between IDs and names (mutually exclusive).
  - IDs must be hard references and can only refer to filters (IP, IPv6, MAC) that already exist.
  - Names can be loose references and can refer to filters (IP, IPv6, MAC) that do not exist.
- *router-name* in mirror destination template
  - Operators can choose between IDs and names (mutually exclusive).
  - IDs must be hard references and can only refer to router IDs (VPRNs) that already exist.
  - Names can be loose references and can refer to router IDs (VPRNs) that do not exist.
- *router-name* in NAT **li-source**
  - Operators can choose between IDs and names (mutually exclusive).
  - IDs must be hard references and can only refer to router IDs (VPRNs) that already exist.
  - Names can be loose references and can refer to router IDs (VPRNs) that do not exist.

[Table 3: Classic CLI engine properties for classic and mixed configuration mode](#) lists the classic CLI engine properties for classic and mixed configuration mode.

*Table 3: Classic CLI engine properties for classic and mixed configuration mode*

Config CLI tree	Classic mode	Mixed mode
<b>li&gt;li-filter-lock-state</b>	<ul style="list-style-type: none"> <li>• locked</li> <li>• unlocked-for-li-users</li> <li>• unlocked-for-all-users</li> </ul>	<ul style="list-style-type: none"> <li>• locked</li> <li>• unlocked-for-all-users</li> </ul> See <a href="#">Configurable filter lock for Lawful Intercept</a> for more information
<b>li&gt;li-source</b>	ID with name	ID with name
<b>li&gt;li-source&gt;nat&gt;classic-lsn-sub&gt;router</b> <b>li&gt;li-source&gt;nat&gt;dslite-lsn-sub&gt;router</b>	ID or name (router and router-name are mutually exclusive)	ID or name (router and router-name are mutually exclusive)



Config CLI tree	Classic mode	Mixed mode
<code>li&gt;li-source&gt;nat&gt;nat64-lsn-sub&gt;router</code>		
<code>li&gt;mirror-dest-template&gt;router</code>	ID or name (router and router-name are mutually exclusive)	ID or name (router and router-name are mutually exclusive)
<code>li&gt;li-filter-associations&gt;li-ip-filter&gt;ip-filter</code> <code>li&gt;li-filter-associations&gt;li-ipv6-filter&gt;ipv6-filter</code> <code>li&gt;li-filter-associations&gt;li-mac-filter&gt;mac-filter</code>	ID or name (ID and name are mutually exclusive)	name only
<code>li&gt;li-filter-block-reservation&gt;li-reserved-block&gt;ip-filter</code> <code>li&gt;li-filter-block-reservation&gt;li-reserved-block&gt;ipv6-filter</code> <code>li&gt;li-filter-block-reservation&gt;li-reserved-block&gt;mac-filter</code>	ID or name (ID and name are mutually exclusive)	name only

Table 4: Reference types for classic and mixed configuration mode lists the reference types for classic and mixed configuration mode.

Table 4: Reference types for classic and mixed configuration mode

Config CLI tree	Classic mode	Mixed mode
<code>li&gt;li-source&gt;sap</code>	loose reference <sup>1</sup>	loose reference
<code>li&gt;li-source&gt;nat&gt;classic-lsn-sub</code>	• hard reference for router <sup>2</sup>	• hard reference for router

<sup>1</sup> Loose reference means that the referenced object does not have to exist before it can be referenced. In addition, the referenced object can be deleted at any time. For example, **li-source** references subscriber-1 even if subscriber-1 is not created on the system. With loose reference, an LI administrator can become completely independent from regular administrators. An LI administrator can provision a list of targets for LI without having to wait for the regular administrator to create them (such as SAP and subscribers). In reverse order, when the regular administrator wants to remove SAPs, there is no need to wait for the LI administrator to remove the reference beforehand. Loose reference allows LI independence and is the recommended model for LI provisioning.

<sup>2</sup> Hard reference means that the object must first be created in the system before it can be referenced. In addition, after an object is referenced by LI, the object must be unreferenced before it can be removed. For example, ip-filter 10 must first exist in the system before it can be referenced using the **li>li-source>ip-filter** command.

Config CLI tree	Classic mode	Mixed mode
<ul style="list-style-type: none"> <li>li&gt;li-source&gt;nat&gt;dslite-ls-sub</li> <li>li&gt;li-source&gt;nat&gt;nat64-ls-sub</li> </ul>	<ul style="list-style-type: none"> <li>• loose reference for router-name</li> </ul> (router and router-name are mutually exclusive)	<ul style="list-style-type: none"> <li>• loose reference for router-name</li> </ul> (router and router-name are mutually exclusive)
<ul style="list-style-type: none"> <li>li&gt;li-source&gt;subscriber</li> <li>li&gt;li-source&gt;nat l2-aware-sub</li> <li>li&gt;li-source&gt;wlan-gw&gt;dsm-subscriber</li> </ul>	loose reference	loose reference
<ul style="list-style-type: none"> <li>li&gt;li-filter-associations&gt;li-ip-filter&gt;ip-filter</li> <li>li&gt;li-filter-associations&gt;li-ipv6-filter&gt;ipv6-filter</li> <li>li&gt;li-filter-associations&gt;li-mac-filter&gt;mac-filter</li> </ul>	hard reference	hard reference
<ul style="list-style-type: none"> <li>li&gt;li-filter-block-reservation&gt;li-reserved-block&gt;ip-filter</li> <li>li&gt;li-filter-block-reservation&gt;li-reserved-block&gt;ipv6-filter</li> <li>li&gt;li-filter-block-reservation&gt;li-reserved-block&gt;mac-filter</li> </ul>	<ul style="list-style-type: none"> <li>• loose reference for filter</li> <li>• hard reference for entries (entries cannot overlap)</li> </ul>	<ul style="list-style-type: none"> <li>• loose reference for filter</li> <li>• hard reference for entries (entries cannot overlap)</li> </ul>
<ul style="list-style-type: none"> <li>li&gt;li-source&gt;ip-filter</li> <li>li&gt;li-source&gt;ipv6-filter</li> <li>li&gt;li-source&gt;mac-filter</li> </ul>	hard reference	not supported
<ul style="list-style-type: none"> <li>li&gt;li-source&gt;li-ip-filter</li> <li>li&gt;li-source&gt;li-ipv6-filter</li> <li>li&gt;li-source&gt;li-mac-filter</li> </ul>	hard reference	hard reference

### 2.2.4.2 LI management using the MD-CLI engine

The MD-CLI engine for mixed and model-driven configuration modes only allows names for filters, router instances, and services; IDs are not supported.

[Table 5: Key differences between classic CLI and MD-CLI](#) lists key differences between classic CLI and MD-CLI.

Table 5: Key differences between classic CLI and MD-CLI

Classic CLI engine for mixed mode	MD-CLI engine for mixed and model-driven mode
<b>li&gt;li-source</b> (ID with name)	<b>/li/li-source</b> (name only)
<b>li&gt;li-source&gt;nat&gt;classic-lsn-sub&gt;router</b> <b>li&gt;li-source&gt;nat&gt;dslite-lsn-sub&gt;router</b> <b>li&gt;li-source&gt;nat&gt;nat64-lsn-sub&gt;router</b> (router ID or name)	<b>/li/li-source/nat/nat44</b> <b>/li/li-source/nat/dslite</b> <b>/li/li-source/nat/nat64</b> (router name only)
<b>li&gt;mirror-dest-template&gt;layer-3-encap&gt;router</b> (router ID or name)	<b>/li/mirror-dest-template/layer-3-encap</b> (router name only)

Compared to classic configuration mode, mixed and model-driven configuration modes primarily use loose references, where the object referenced does not have to exist in the system before it is referenced. For example, subscriber-1 is referenced in **li-source** but does not need to be created on the system beforehand.

In classic configuration mode, when an LI filter (**li-ip-filter**, **li-ipv6-filter**, and **li-mac-filter**) is configured:

- LI filter entries can be referenced even if the LI filter is not associated (under the **config>li>li-filter-associations** context) with a filter in the **config>filter** context.
- After an LI filter entry is referenced in the **li-source**, the LI filter association that contains the specified LI filter cannot be deleted. Therefore, the LI filter association can only be deleted if the LI filter entries are no longer referenced in the **li-source**.
- When an LI filter entry is not referenced in **li-source**:
  - If **li-filter-associations** associates an LI filter name with a filter ID, the deletion of either the filter ID or the LI filter name, also deletes the **li-filter-associations**.
  - After **li-filter-associations** associates an LI filter name with filter name, the deletion of either the LI filter name or the filter name is always denied and no roll back to a configuration mode is completed without the LI filter name or the filter name.

In mixed configuration mode:

- An LI filter entry can be referenced in **li-source**, when the LI filter is first associated with a filter in the **config>filter** context under the **config>li>li-filter-associations** context.
- LI filter association between an LI filter name and a filter ID is not allowed.
- After an LI filter entry is referenced in the **li-source**, the LI filter association that contains the specified LI filter cannot be deleted. Therefore, the LI filter association can only be deleted if the LI filter entries are no longer referenced in the **li-source**.
- When an LI filter entry is not referenced in **li-source**; after **li-filter-associations** associates an LI filter name with filter name, the deletion of either the LI filter name or the filter name is always denied and no roll back to a configuration mode is completed without the LI filter name or the filter name.

### 2.2.4.3 Lawful Intercept in NETCONF

LI can be managed using NETCONF.

The LI configuration is located in separate data stores that are distinct from the rest of the general configuration data.

See "NETCONF" in the *7450 ESS, 7750 SR, 7950 XRS, and VSR System Management Guide* for more information about LI and NETCONF.

### 2.2.4.4 Lawful Intercept in gNMI

The default AAA profiles do not block SET, GET, or SUBSCRIBE access to LI state data. Creating a new AAA profile is recommended to control gNMI user access to LI state data.

### 2.2.4.5 Mixed mode SNMPv3 support

When mixed mode is enabled, a limited set of Lawful Intercept (LI) commands can be executed with SNMPv3. The following commands are supported in mixed mode for SNMPv3:

- **MD-CLI**

```
admin save li
li li-source sap
li li-source subscriber
```

- **classic CLI**

```
configure li save
configure li li-source sap
configure li li-source subscriber
```

### 2.2.4.6 CLI configuration mode migration

Lawful Intercept can be managed in classic, mixed, or model-driven configuration mode.



**Note:** The LI administrator should coordinate configuration mode migrations with the network administrator, who is normally expected to perform the configuration mode migrations.

In LI there are two configuration modes of operations dictated by using the **bof li-separate** command:

- When **li-separate** is enabled, the LI management is controlled by the LI administrator who is given "access li" rights to access the LI region. It is highly recommended to consider adjusting the member profile. See [Mandatory LI profile migration](#) for information about profiles.
- When **li-separate** is disabled (**no li-separate**), "access li" no longer determines who has access to LI. Instead, access to the LI region is governed by the "AAA profile" (MD-CLI) or "profile" (Classic CLI) applied against the user. Inside the "AAA profile" there are CLI filters, for example, **configure**, **show**, and **clear**, to allow or deny which CLI commands can be accessed. It is highly recommended to consider the following when operating with **no li-separate**.
  - A good practice is to add "access li" to users who require access to LI.
  - See [Mandatory LI profile migration](#) for information about profiles.



**Note:** When performing a configuration mode migration from classic to mixed or model-driven configuration mode, migration steps may be required (see [Configuration mode migration check list](#)). Migrating from mixed or model-driven configuration mode to classic configuration mode does not require any migration steps.

### 2.2.4.6.1 Configuration mode migration check list

When performing a configuration mode change from classic to mixed or model-driven configuration mode, operators are highly recommended to perform the following migration procedures:

- LI profile migration
- configuration migration
- how to complete the configuration mode migration

#### 2.2.4.6.1.1 Mandatory LI profile migration

LI administrators must update the profile for model-driven configuration access to the LI region. Without the update, the LI administrator cannot provision LI in MD-CLI.

This step must be performed before a configuration mode migration from classic to mixed or model-driven configuration mode. The existing profile for LI under the **config>system>security>profile** context can only provide LI access to the LI administrator or the LI users for the classic CLI engine.



**Note:** The "li access" profile is not a default profile created by SR OS. It is a profile created by the administrator. Search for entries with **configure li**, **show li**, **admin save li**, and **clear li** inside created profiles. A profile that allows LI access typically allows these commands. It is highly recommended that only users who have access rights to LI apply the LI profile. It is also highly recommended that all other profiles deny **configure li**, **show li**, **admin save li**, and **clear li** commands for all other users.

Profiles are not automatically updated for MD-CLI commands. The administrator is responsible for creating an LI filter list for the MD-CLI that is equivalent to the classic CLI. This is highly recommended for the **li-separate** and **no li-separate** commands. This step must be performed before the configuration mode migration.

The existing profile for LI access should, at a minimum, include the following:

```
config>system>security>profile
li
entry n
  match "configure li"
  action permit
```

At minimum, add the following MD-CLI commands to the existing LI profile that grants user access to LI commands:

```
entry n
  match "li"
  action permit
entry n+1
  match "edit-config li"
  action permit
entry n+2
```

```

match "admin save li"
  action permit
entry n+3
  match "commit"
  action permit
entry n+4
  match "compare"
  action permit
entry n+5
  match "tools perform management-interface configuration-mode"
  action permit
entry n+6
  match "quit-config li"
  action permit
entry n+7
  match "state li"
  action permit

```

It is recommended to block the following access for all other users. This is accomplished either through **default-action deny** or through explicit **deny** commands. The following are the recommended MD-CLI commands that deny access to specific users:

```

entry n
  match "li"
  action deny
entry n+1
  match "edit-config li"
  action deny
entry n+2
  match "admin save li"
  action deny
entry n+3
  match "state li"
  action deny

```

### 2.2.4.6.1.2 Configuration mode migration

Switching from classic configuration mode to mixed or model-driven configuration mode is normally performed by the network administrator using the **configure system mgmt-itf configuration-mode [mixed | model-driven]** command. The LI administrator can use the **tools perform system management-interface configuration-mode [mixed | model-driven] check li** command to test the LI configuration for a configuration mode migration. Only the LI administrator can use the tools command. When **li-separate** is not set, the user must have access to the LI and this is determined by the user profile. Within the user profile, the user must have access to **configure li** CLI commands. When **li-separate** is set, the user must have **access li** and also have **configure li** included in the user profile.

```

config>system>security>profile
  li
entry n
  match "configure li"
  action permit

```

If the network administrator attempts to perform the configuration mode migration and the LI configuration requires migration, the following message appears:

Action required: LI configuration requires updating before configuration mode switch

To track details of the LI migration steps for a configuration mode migration, the LI administrator's configuration must include "access li". The LI administrator can then use the **tools perform system mgmt-itf>configuration-mode [mixed | model-driven] check li** command. This command serves two purposes:

- verifies the steps necessary for the configuration mode migration
- ensures all configuration can be migrated

The LI administrator should follow the instruction returned by the **tools** command to prepare the LI configuration for migration. See [Migrating from classic to mixed or model-driven configuration mode](#) for information about the list of migration steps. After completing the migration steps, the network administrator can execute the **config system mgmt-itf configuration-mode [mixed | model-driven]** command, and then the configuration mode migration immediately takes effect.

### 2.2.4.6.1.3 Configuration mode migration completion

If the **li-local-save** command is enabled on the BOF, saving the LI configuration is highly recommended after every configuration mode change.

The main configuration file (default: `config.cfg`) determines the system bootup configuration mode, specifically from the command line **configure system management-interface configuration-mode**. When the administrator executes the configuration mode change, the LI configuration file format remains as the last saved format until an **admin save li** command is executed. For example, when migrating from classic to model-driven configuration mode, the LI configuration in the file `li.cfg` remains in the classic format until the **admin save li** command is executed to update the file format to a model-driven format.



**Note:** Failing to execute the **admin save li** command after a configuration mode migration may cause LI configuration bootup failure.

If the LI configuration fails to boot because the **admin save li** command is not performed immediately after a configuration mode migration, both log 99 and console dump inform the LI configuration field to load because the LI format does not match the primary configuration format file. The format of both the `li.cfg` file and the main configuration must match. To recover, the main configuration must be rolled back to the previous configuration mode to match the `li.cfg` file saved format (as indicated by console dump or log 99), and then rebooted.

It is important not to perform a **save li** command when there is a configuration mode mismatch between the main configuration (default `config.cfg`) file and the `li.cfg` file. Saving the `li.cfg` file creates a new file without any configuration. The previously generated `li.cfg` file is archived as `li.cfg.1` file. If a **save li** command is accidentally executed, perform the following steps:

1. Roll back to the previous configuration mode in which the `li.cfg.1` file is saved (as indicated by console dump or log 99).
2. Reboot the system.
3. Restore the `li.cfg.1` file using the **exec li.cfg.1** command.

### 2.2.4.6.1.4 Migrating from classic to mixed or model-driven configuration mode

The following are the migration procedures necessary to migrate from classic configuration mode to mixed or model-driven configuration mode:

- **config li li-filter-block-reservation li-reserved-block**

If the filter uses IDs, then all IDs must be referenced in an existing filter ID. All loose references (filter IDs that do not exist in the main configuration region), must be removed. This does not impact any existing LI services because unreferenced filters are not in use.

If the filter uses names, no migration step is needed.

- **config li li-filter-lock-state**

If the state is configured with **unlocked-for-li-users**, this must be changed to **unlocked-for-all-users** or **locked**. See [Configurable filter lock for Lawful Intercept](#) for information about lock states. Changing the lock state does not impact the LI services.

- **configure>li>li-source>[{ip-filter | ipv6-filter | mac-filter}]**

If any of these filters must be removed, an LI-based filter (**li-ip-filter**, **li-ipv6-filter**, **li-mac-filter**) can be used in place of direct referenced filters. It is possible for the LI-based filter to reference the same reference filter. For example, **li-source** has **ip-filter** 1 entry 1 applied. It is possible to have **li-ip-filter** "one" associated with **ip-filter** 1, and then apply **li-ip-filter** "one" entry 1 to **li-source**. Then remove the **ip-filter** from **li-source** and remove the entry from the **ip-filter** 1 to make **li-ip-filter** effective. This migration can help minimize the disruption to the LI service.

### 2.2.4.6.1.5 Further information and recommendations on LI

Further recommendations and more information about LI are as follows:

- In mixed configuration mode, all users (LI and non-LI users) should only use private candidate for configuration, to allow classic interfaces such as SNMP to manage the router.
- Other model-driven interfaces such as NETCONF can perform an exclusive lock on the LI configuration. This prevents direct CLI configuration and therefore, the best way to configure LI is to use a single model-driven interface at a time.
- Command completion of parameter values between the LI and configure regions is not supported in the MD-CLI.
- When the **li-separate** command is enabled, the MD-CLI does not support the **load** or **rollback** commands in the LI and configure regions.
- The LI configuration is not saved to the startup configuration file when changes are committed. Configuration changes must be manually saved with the MD-CLI **admin save configuration li** command.
- If the LI configuration file fails to load at boot, the **admin show configuration li** command can be used to view the LI configuration file by the LI administrator.

## 2.2.5 Configuring Lawful Intercept in model-driven interface

The MD-CLI supports two configuration work flows for LI:

- **implicit configuration work flow**
  - Navigation is restricted to the **li** branch and its descendants.
  - Operational commands require an absolute path and error when incomplete.
  - The **li {private | exclusive | read-only}** command enters the configuration mode and navigates in the **li** branch. There is no default configuration mode.



- The **exit all** command leaves the configuration mode and navigates to the operational root.
- **explicit configuration work flow**
  - Navigation is unrestricted while in configuration mode.
  - Operational commands while in the **li** branch require an absolute path and navigate when incomplete.
  - The **edit-config li {private | exclusive | read-only}** command enters the configuration mode without navigating. There is no default configuration mode.
  - The **quit-config** command leaves the configuration mode without navigating. The **quit-config** command is not available in the **configure** branch.

See the *7450 ESS, 7750 SR, 7950 XRS, and VSR MD-CLI Quick Reference Guide* and the *7450 ESS, 7750 SR, 7950 XRS, and VSR MD-CLI Command Reference Guide* for more information about data stores, transactions, candidates, and using configuration commands in the MD-CLI.

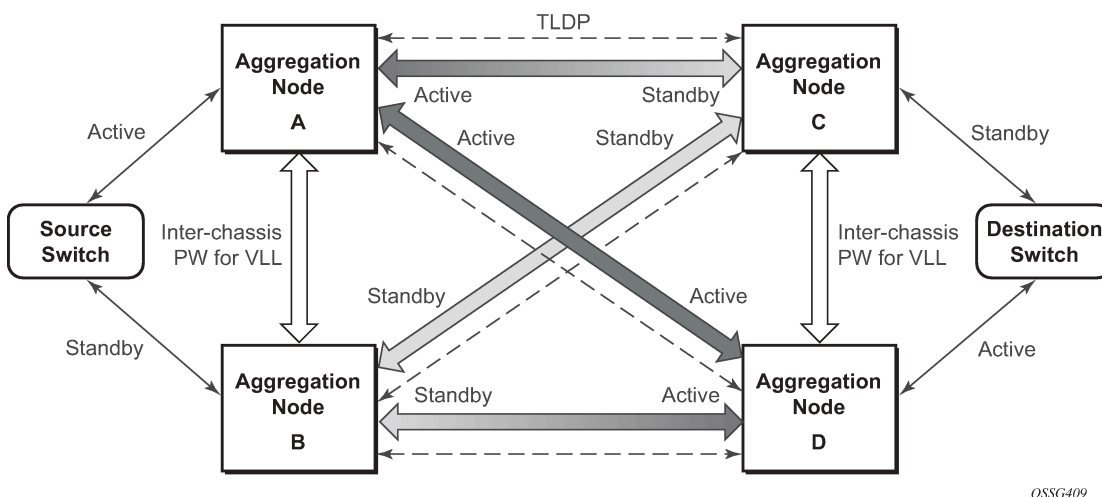
For LI configuration over NETCONF information, see the *Datastores and URLs and NETCONF Operations and Capabilities* sections in the *7450 ESS, 7750 SR, 7950 XRS, and VSR System Management Guide*.

## 2.3 Pseudowire redundant mirror services

This section describes the implementation and configuration of redundant Mirror/Lawful Intercept services using redundant pseudowires.

Regardless of the protection mechanism (MC-LAG, STP, or APS) the source switch only transmits on the active link and not simultaneously on the standby link. As a result, when configuring a redundant mirror or LI service or a mirror service where the customer has a redundant service but the mirror or LI service is not redundant the mirror source must be configured on both (A and B) PE nodes. In either case, the PE with a mirror source establishes a pseudowire to each eligible PE where the mirror / LI service terminates.

Figure 10: State engine for redundant service to a redundant mirror service

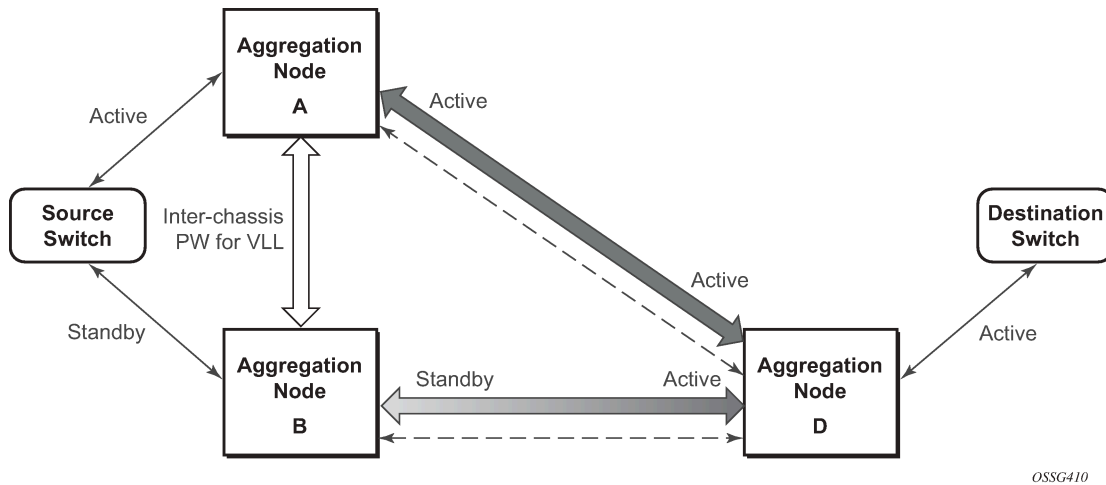


It is important to note that to provide protection if the active SDP between node A and D fails and the need to limit the number of lost data for LI the ICB between node A and B must be supported. As a result, when

the SDP connecting nodes A and D fails the data on its way from the source switch to node A and the data in node A must be directed by the ICB to node B and from there to node D.

This functionality is already supported in when providing pseudo wire redundancy for VLLs and must be extended to mirror or LI service redundancy.

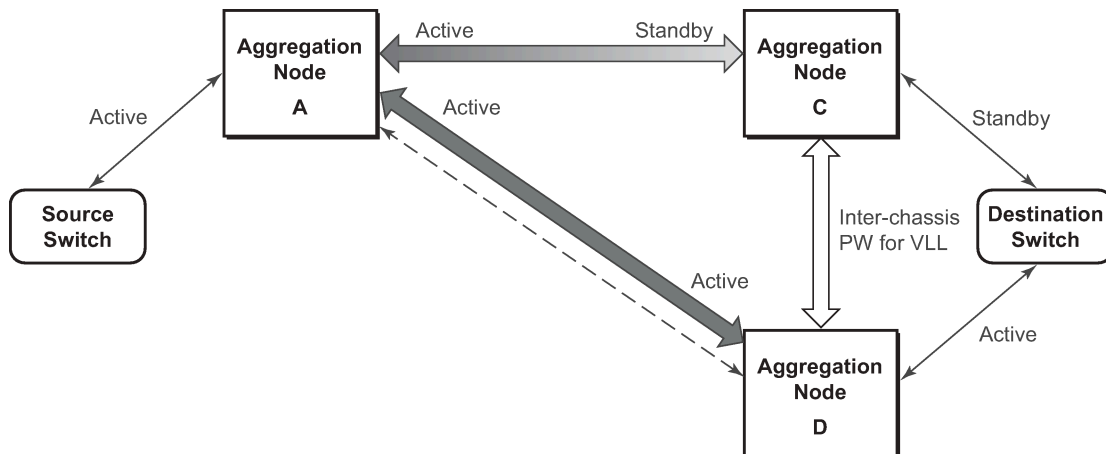
Figure 11: State engine for redundant service to a non-redundant mirror service



OSSG410

The notable difference with scenarios standard pseudo wire redundancy scenarios is that provided the customer service is redundant on nodes A and B (Figure 10: State engine for redundant service to a redundant mirror service and Figure 11: State engine for redundant service to a non-redundant mirror service) both aggregation node A and Aggregation node B maintain an active Pseudo wire to Node D who in turn has an active link to the destination switch. If in Figure 10: State engine for redundant service to a redundant mirror service, the link between D and the destination switch is disconnected, then both aggregation A and B must switch to use pseudowire connection to Node C.

Figure 12: State engine for a non-redundant service to a redundant mirror service



OSSG411

In the case where a non-redundant service is being mirrored to a redundant mirror service (Figure 12: State engine for a non-redundant service to a redundant mirror service ) the source aggregation node (A) can only maintain a pseudo wire to the active destination aggregation node (D). Should the link between

aggregation node D and the destination switch fail then the pseudo wire must switch to the new active aggregation node (C).

### 2.3.1 Redundant mirror source notes

A redundant remote mirror service destination is not supported for IP mirrors (a set of remote IP mirror destinations). The remote destination of an IP mirror is a VPRN instance, and an "endpoint" cannot be configured in a VPRN service.

A redundant mirror source is supported for IP mirrors, but the remote destination must be a single node (a set of mirror source nodes, each with a mirror destination that points to the same destination node). In this case the destination node would have a VPRN instance with multiple ip-mirror-interfaces.

Multi Chassis APS (MC-APS) groups cannot be used as the SAP for a redundant remote mirror destination service. APS cannot be used to connect the remote mirror destination SR nodes to a destination switch.

Multi Chassis APS (MC-APS) groups can be used as the SAP for a redundant mirror service source. APS can be used to redundantly connect the source of the mirrored traffic to the SR nodes that are behaving as the mirror-sources.

## 2.4 Configuration process overview

Mirroring can be performed based on the following criteria:

- [Port](#)
- [SAP](#)
- [MAC filter](#)
- [IP filter](#)
- [Ingress label](#)
- [Subscriber](#)

Configuring mirroring is like creating a unidirection service. Mirroring requires the configuration of:

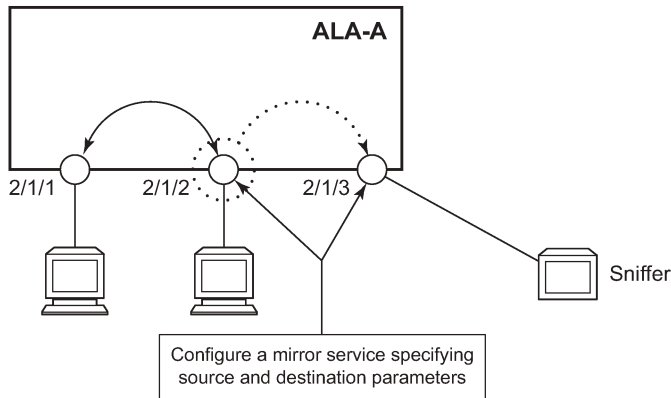
**mirror source**                    the traffic on specific points to mirror

**mirror destination**            the location to send the mirrored traffic, where the sniffer is to be located

[Figure 13: Local mirroring example](#) shows a local mirror service configured on ALA-A.

- Port 2/1/2 is specified as the source. Mirrored traffic ingressing and egressing this port is sent to port 2/1/3.
- SAP 2/1/3 is specified as the destination. The sniffer is physically connected to this port. Mirrored traffic ingressing and egressing port 2/1/2 is sent here. SAP, encapsulation requirements, packet slicing, and mirror classification parameters are configured. SDPs are not used in local mirroring.

Figure 13: Local mirroring example

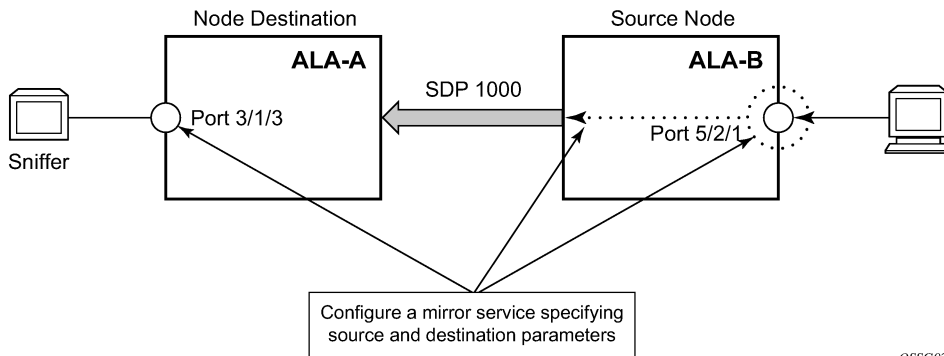


OSSG026

Figure 14: Remote mirroring example shows a remote mirror service configured as ALA B as the mirror source and ALA A as the mirror destination. Mirrored traffic ingressing and egressing port 5/2/1 (the source) on ALA B is handled the following ways:

- Port 5/2/1 is specified as the mirror source port. Parameters are defined to select specific traffic ingressing and egressing this port.
- Destination parameters are defined to specify where the mirrored traffic is to be sent. In this case, mirrored traffic is sent to a SAP configured as part of the mirror service on port 3/1/3 on ALA A (the mirror destination).
- ALA A decodes the service ID and sends the traffic out of port 3/1/3.
- The sniffer is physically connected to this port (3/1/3). SAP, encapsulation requirements, packet slicing, and mirror classification parameters are configured in the destination parameters.

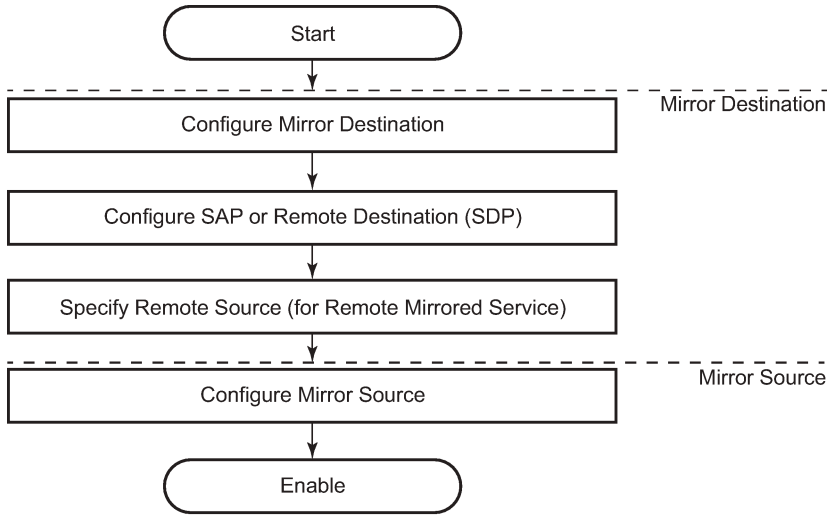
Figure 14: Remote mirroring example



OSSG027

Figure 15: Mirror configuration and implementation flow shows the process to provision basic mirroring parameters.

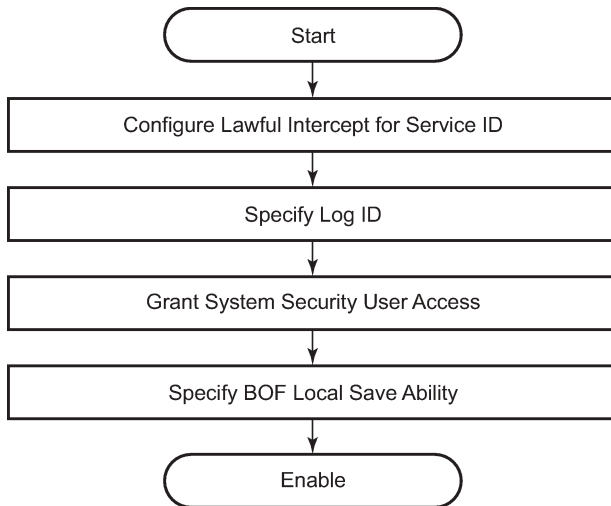
Figure 15: Mirror configuration and implementation flow



OAM\_14

Figure 16: Lawful intercept configuration and implementation flow shows the process to provision LI parameters.

Figure 16: Lawful intercept configuration and implementation flow



OAM\_15

## 2.5 Configuration notes

This section describes mirroring configuration restrictions:

- Multiple mirroring service IDs (mirror destinations) may be created within a single system.
- A mirrored source can only have one destination.

- Both destination mirroring service IDs (including service parameters) and config mirror source (defined in **config>mirror>mirror-source**) are persistent between router (re)boots and are included in the configuration saves.

Debug mirror source (defined **debug>mirror>mirror-source**) and lawful intercept source (defined in **config>li>li-source**) criteria configurations are not preserved in a configuration save (**admin save**). Debug mirror source configuration can be saved using **admin>debug-save**. Lawful intercept source configuration can be saved using **config>li>save**.

- Subscriber based lawful intercept source criteria is persistent across creation/existence of the subscriber. Filter or SAP-based LI source criteria is removed from the LI source configuration if the filter entry or SAP is deleted. Applies to the 7450 ESS and 7950 SR.
- Physical layer problems such as collisions, jabbers, and so on, are not mirrored. Typically, only complete packets are mirrored.
- Starting and shutting down mirroring:

#### **mirror destinations**

- The default state for a mirror destination service ID is shutdown. Execute a **no shutdown** command to enable the feature.
- When a mirror destination service ID is shutdown, mirrored packets associated with the service ID are not accepted from its mirror source or remote source. The associated mirror source is put into an operationally down mode. Mirrored packets are not transmitted out the SAP or SDP. Each mirrored packet is silently discarded. If the mirror destination is a SAP, the SAP's discard counters are incremented.
- Issuing the shutdown command causes the mirror destination service or its mirror source to be put into an administratively down state. Mirror destination service IDs must be shut down first to delete a service ID, or SAP, or SDP association from the system.

#### **mirror sources**

- The default state for a mirror source for a mirror-dest service ID is **no shutdown**. Enter a **shutdown** command to deactivate (disable) mirroring from that mirror-source.
- Mirror sources do not need to be shutdown to remove them from the system. When a mirror source is shutdown, mirroring is terminated for all sources defined locally for the mirror destination service ID.

The following are lawful intercept configuration restrictions.

To address network management, operators without LI permission cannot view or manage the LI data on the node nor can they view or manage the data on the Network Management platform.

Entries within LI filters (**li-ip-filter**, **li-ipv6-filter**, and **li-mac-filter**) are typically used to match a specific IP or MAC address as LI targets. When these LI filters are associated with a filter in the main configuration region (**ip-filter**, **ipv6-filer**, or **mac-filter**), the system does not insert the entries in a sequence for performance reasons. For example, it is possible that filter entry 2 can take place before filter entry 1. This does not affect the LI functionality. However, in cases where the entries overlap, it is possible for a latter entry to first match the IP or the MAC address.

LI mirroring does not allow the configuration of ports and ingress labels as a source parameter.

## 2.6 Configuring service mirroring with CLI

This section provides information about service mirroring.

### 2.6.1 Mirror configuration overview

SR OS mirroring can be organized in the following logical entities:

- The mirror source is defined as the location where ingress or egress traffic specific to a port, SAP, MAC, or IP filter, ingress label or a subscriber is to be mirrored (copied). The original frames are not altered or affected in any way.
- An SDP is used to define the mirror destination on the source router to point to a remote destination (another router).
- A SAP is defined in local and remote mirror services as the mirror destination to where the mirrored packets are sent.
- The subscriber contains hosts which are added to a mirroring service (applies to the 7450 SR and 7750 SR only).

#### 2.6.1.1 Defining mirrored traffic

In some scenarios, like using VPN services or when multiple services are configured on the same port, specifying the port does not provide a sufficient resolution to separate traffic. In Nokia's implementation of mirroring, multiple source mirroring parameters can be specified to further identify traffic.

Mirroring of packets matching specific filter entries in an IP or MAC filter can be applied to refine what traffic is mirrored to flows of traffic within a service. The IP criteria can be combinations of:

- source IP address and mask
- destination IP address and mask
- IP protocol value
- source port value and range (for example, UDP, or TCP port)
- destination port value and range (for example, UDP, or TCP port)
- DiffServ Code Point (DSCP) value
- ICMP code
- ICMP type
- IP fragments
- IP option value and mask
- single or multiple IP option fields present
- IP option fields present
- TCP ACK set/reset
- TCP SYN set/reset
- SAP ingress/egress labels

The MAC criteria can be combinations of:

- IEEE 802.1p value and mask
- source MAC address and mask
- destination MAC address and mask
- Ethernet Type II Ethernet type value
- Ethernet 802.2 LLC DSAP value and mask
- Ethernet 802.2 LLC SSAP value and mask
- IEEE 802.3 LLC SNAP Ethernet frame OUI zero or non-zero value
- IEEE 802.3 LLC SNAP Ethernet frame PID value
- SAP ingress/egress labels

## 2.6.2 Lawful Intercept configuration overview

Lawful Intercept allows the user to access and execute commands at various command levels based on profiles assigned to the user by the administrator. LI must be configured in the **config>system>security>user>access** and **config>system>security>profile** contexts. The options include FTP, SNMP, console, and LI access.

LI parameters configured in the BOF context (**li-local-save** and **li-separate**) include the ability to access LI separately than the normal administrator. As with all BOF entities, changing the BOF file during normal system operation only results in the parameter being set for the next reboot. These BOF commands are initialized to the default values, **no li-separate** and **no-li-local-save**. A system boot is necessary for any change to the **li-separate** and **li-local-save** to become effective.

Changes to the **li-separate** and **li-local-save** configurations should be made in both primary and backup CM BOF files.

At regular intervals, a LI status event is generated by the system to indicate the mode of the LI administration, time of the last reboot, and whether local save is enabled.

### 2.6.2.1 Saving LI data

Depending on location and law enforcement preferences, the node can be configured to save all LI data on local media. If the operator saves this data then when starting or restarting the system the configuration file is processed first and then the LI configuration is restarted.

When permitted to save the data, the data is encrypted using AES. The encryption key is unique per system and is not visible to any administrator.

To save LI data locally, the option must be configured in the **bof>li-local-save** context. Enabling this option is only applied after a system reboot.

If an LI save is permitted, then only a local save is permitted and, by default, it is saved to Compact Flash 3 with the filename of `li.cfg`. An explicit save command under the **config>li** context must be executed to save the LI. An LI administrator with privileges to configure LI, can execute the `li.cfg` file.

### 2.6.2.2 Regulating LI access

Depending on local regulations pertaining to Lawful Intercept (LI) a node can be configured to separate normal system administration tasks from tasks of a Lawful Intercept operator.



If the separation of access is not required and any administrator can manage lawful intercept or plain mirroring, then it is not necessary to configure the **li-separate** parameter in the BOF configuration. However, to ensure logical separation, the following must occur:

1. An administrator must create a user and configure the user as LI capable (**config>system>security>user>access** context). Furthermore, the administrator must assure that both CLI and SNMP access permission is granted for the LI operator.
2. Finally, before turning the system into two separate administration domains, the CLI user must be granted a profile that limits the LI operator to those tasks relevant to the job (**config>system>security>profile>li** context).

It is important to remember that the LI operator is the only entity who can grant LI permission to any other user when in **li-separate** mode.

Provided the above procedure is followed, the LI administrator must decide whether to allow the LI (source) configuration to be saved onto local media. This is also subject to local regulations.

At this point, the BOF file can be configured with the **li-separate** and **li-local-save** parameters. If the local save is not configured then the LI information must be reconfigured after a system reboot.

Assuming **li-separate** is configured, the node should be rebooted to activate the **separate** mode. At this point the system administrators without LI permission cannot modify, create or view any LI- specific configurations. For this to occur, the BOF file must be reconfigured and the system rebooted. This combined with other features prohibits an unauthorized operator from modifying the administrative separation without notifying the LI administrator.

The following example shows an SNMP configuration with views, access groups, and attempts parameters.

```
A:ALA-23>config>system>security>snmp# info detail
-----
      view iso subtree 1
            mask ff type included
      exit
      view no-security subtree 1
            mask ff type included
      exit
      view no-security subtree 1.3.6.1.6.3
            mask ff type excluded
      exit
      view no-security subtree 1.3.6.1.6.3.10.2.1
            mask ff type included
      exit
      view no-security subtree 1.3.6.1.6.3.11.2.1
            mask ff type included
      exit
      view no-security subtree 1.3.6.1.6.3.15.1.1
            mask ff type included
      exit
      ...
      access group "snmp-li-ro" security-model usm security-
level privacy context "li" read "li-view" notify "iso"
      access group "snmp-li-rw" security-model usm security-
level privacy context "li" read "li-view" write "li-view" notify "iso"
      attempts 20 time 5 lockout 10
      ...
-----
A:ALA-23>config>system>security>snmp#
```

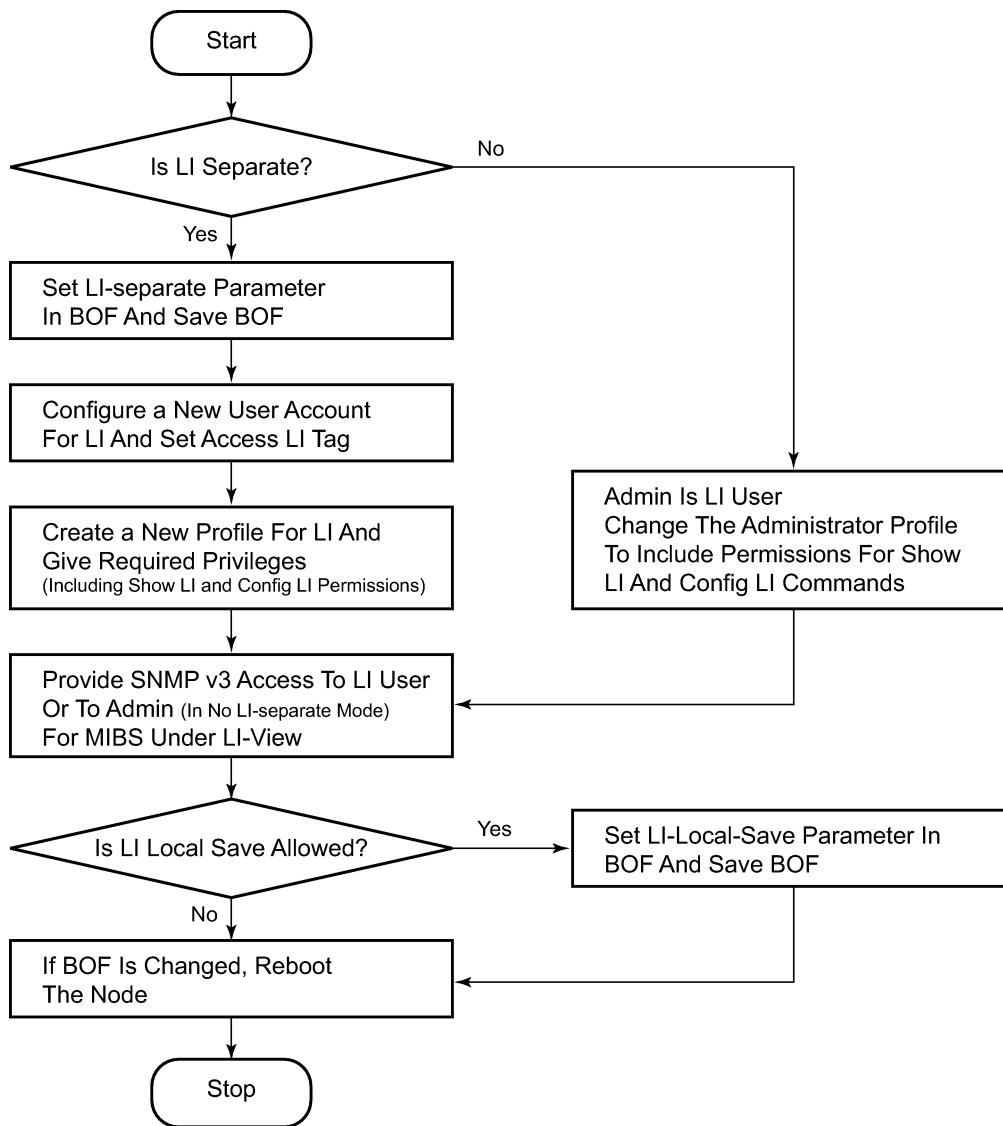
The following example shows a user account configuration.

```
A:ALA-23>config>system>security# info
-----
...
  user "liuser"
    access console snmp li
    console
      no member "default"
      member "liprofile"
    exit
  snmp
    authentication md5 <auth-key> privacy des <priv-key>
    group "snmp-li-rw"
  exit
exit
...
-----
A:ALA-23>config>system>security#
```

### 2.6.2.2.1 LI user access

By default, LI user access is limited to those commands that are required to manage LI functionality. If a user is granted permission to access other configuration and operational data, then this must be explicitly configured in the user profile of the LI operator in the **config>system>security>profile>entry>match command-string** context. [Figure 17: Creating an LI operator account](#) shows the work flow to set an LI operator.

Figure 17: Creating an LI operator account



OSSG264

### 2.6.2.2.2 LI source configuration

Filter configuration is accessible to both the LI operator and regular system administrators. If the content of a filter list that is subject to an LI operation and if a filter (included in the filter list) is used by an LI operator, its contents cannot be modified unless the **li-filter-lock-state** is unlocked, see [Configurable filter lock for Lawful Intercept](#). If an attempt is made, then an LI event is generated. An LI source can contain many LI filter entries. In general, an LI source can only associate with one mirror destination service. A mirror destination can be associated with one source: debug mirror source, config mirror source, or LI mirror source. When a mirror destination is referenced by a source, the mirror destination cannot be referenced again.

In the configuration, when an LI operator specifies that an entry must be used as an LI entry then this fact is hidden from all non-LI operators. Modification of a filter entry is not allowed if it is used by LI, see [Configurable filter lock for Lawful Intercept](#). However, an event is generated, directed to the LI operator, indicating that the filter has been compromised.

Debug mirroring source has the lowest priority compared to both **config mirror source** and **li source**, for example, when a SAP is referenced in a debug mirror source. It is possible for the **config mirror source** or **li source** to reference the same SAP. The debug mirror source SAP is silently deleted.

The following order applies for both ingress and egress traffic:

- port mirroring (debug only)
- SAP mirroring (debug or LI)
- subscriber mirroring (debug or LI) for the 7450 ESS and 7750 SR
- filter mirroring (debug or LI)

For frames from network ports:

- port mirroring (debug only)
- label mirroring (debug only, ingress only)
- filter mirroring (debug or LI)

Filters can be created by all users that have access to the relevant CLI branches.

When an LI mirror source using a specific service ID is created and is in the **no shutdown** state, the corresponding mirror destination on the node cannot be modified (including **shutdown/no shutdown** commands) or deleted.

In the **separate** mode, the anonymity of the source is protected. After source criterion is attached to the LI source, the following applies:

- In SAP configurations, only modifications that stop the flow of LI data while the customer receives data is blocked unless the **li-filter-lock-state** is unlocked, see [Configurable filter lock for Lawful Intercept](#).
- In filter configurations, if a filter entry is attached to the LI source, modification and deletion of both the filter and the filter entry are blocked.

### 2.6.2.3 Configurable filter lock for Lawful Intercept

With the default Lawful Intercept configuration, when a filter entry is used as a Lawful Intercept (LI) mirror source criteria/entry, all subsequent attempts to modify the filter are then blocked to avoid having the LI session impacted by a non-LI user.

A configurable LI parameter allows an LI user to control the behavior of filters when they are used for LI.

Configuration of the **li-filter-lock-state** allows an operator to control whether modifications to filters that are being used for LI are allowed by no users, all users or li users only.

### 2.6.2.4 LI MAC filter configuration

Although normal MAC filter entries (configured under **config>filter>mac-filter**) can be referenced in an **li-source**, there is also the option to configure and use special-purpose Lawful Intercept MAC filters.

LI MAC filters are configured in the protected **config>li** CLI branch.

LI MAC filters are associated by configuration with normal MAC filters, and entries created in the LI MAC filters are inserted into the associated normal MAC filter before the filter is downloaded to the data plane hardware and applied. The combined filter list is not visible to any users which maintains a separation between LI operators and operators doing other normal filter configuration work (for example, interface ACLs).

A configurable **li-filter-block-reservation** is used to reserve a range of entries in the normal filter into which the LI entries are inserted.

### 2.6.2.5 LI logging

A logging collector is supported in addition to existing main, security, change, and debug log collectors. LI log features include the following:

- Only visible to LI operators (such as show command output).
- Encrypted when transmitted (SNMPv3).
- Logging ability can only be created, modified, or deleted by an LI operator.
- The LI user profile must include the ability to manage the LI functions.

### 2.6.3 Basic mirroring configuration

Destination mirroring parameters must include at least:

- a mirror destination ID (same as the mirror source service ID)
- a mirror destination SAP or SDP

Mirror source parameters must include at least:

- a mirror service ID (same as the mirror destination service ID)
- one source type (port, SAP, ingress label, IP filter or MAC filter) specified

The following example shows a configuration of a local mirrored service where the source and destinations are on the same device (ALA-A).

```
*A:ALA-A>config>mirror# info
-----
      mirror-dest 103 create
      sap 2/1/25:0 create
egress
      qos 1
      exit
      exit
      no shutdown
      exit
-----
*A:ALA-A>config>mirror#
```

The following examples shows a mirror source configuration:

```
*A:ALA-A>debug>mirror-source# show debug mirror
debug
      mirror-source 103
      port 2/1/24 egress ingress
no shutdown
```

```

    exit
  exit
  *A:ALA-A>debug>mirror-source# exit

```

The following example shows a configuration of a remote mirrored service where the source is a port on ALA-A and the destination is a SAP is on ALA-B:

```

*A:ALA-A>config>mirror# info
-----
    mirror-dest 1000 create
      spoke-sdp 2:1 egr-svc-label 7000
      no shutdown
    exit
-----
*A:ALA-A>config>mirror# exit all
*A:ALA-A# show debug
debug
  mirror-source 1000
    port 2/1/2 egress ingress
  no shutdown
  exit
exit
*A:ALA-A#

*A:ALA-B>config>mirror# info
-----
    mirror-dest 1000 create
      remote-source
        far-end 10.10.10.104 ing-svc-label 7000
      exit
    sap 3/1/2:0 create
egress
  qos 1
  exit
  exit
  no shutdown
  exit
-----
*A:ALA-B>config>mirror#

```

### 2.6.3.1 Mirror classification rules

Nokia's implementation of mirroring can be performed by configuring parameters to select network traffic according to any of the following entities.

#### 2.6.3.1.1 Port

The port command associates a port to a mirror source. The port is identified by the port ID.

The following shows the *port-id* syntax for the **port** command:

Table 6:

<i>port-id:</i>	<b>slot/mda/port[.channel]</b>	
eth-sat-id	<b>esat-id/slot/port</b>	
	<b>esat</b>	keyword
	<i>id</i>	1 to 20
pxc-id	<b>pxc-id.sub-port</b>	
	<b>pxc</b>	keyword
	<i>id</i>	1 to 64
	<i>sub-port</i>	a, b
	<b>ccag-id - ccag-id.path-id[cc-type]:cc-id</b>	
	<b>ccag</b>	keyword
	<i>id</i>	1 to 8
	<i>path-id</i>	a,b
	<i>cc-type</i>	.sap-net, .net-sap
	<i>cc-id</i>	0 to 4094
lag-id	1 to 800	
<b>egress</b>	keyword	
<b>ingress</b>	keyword	



**Note:** On the 7950 XRS, the XMA ID takes the place of the MDA.

The defined port can be an Ethernet port, a SONET/SDH path, a TDM channel, a Cross Connect Aggregation Group (CCAG), or a Link Aggregation Group (LAG) ID. If the port is a SONET/SDH or TDM channel, the channel ID must be specified to identify which channel is being mirrored. When a LAG ID is specified as the port ID, mirroring is enabled on all ports making up the LAG. Ports that are circuit-emulation (CEM) cannot be used in a mirror source (applies to the 7750 SR). SONET/SDH applies to the 7450 ESS and 7750 SR, and TDM channel apply to the 7750 SR.

Mirror sources can be ports in either access or network mode. Port mirroring is supported in the following combinations:

Table 7: Mirror source port requirements

Port type	Port mode	Port encapsulation type
faste/gige/xgige Ethernet	access	dot1q, null, QinQ

Port type	Port mode	Port encapsulation type
faste/gige/xgige Ethernet	network	dot1q, null

```
debug>mirror-source# port {port-id | lag lag-id}
{[egress][ingress]}
```

```
*A:ALA-A>debug>mirror-source# port 2/2/2 ingress egress
```

### 2.6.3.1.2 SAP

More than one SAP can be associated within a single mirror-source. Each SAP has its own ingress and egress parameter keywords to define which packets are mirrored to the mirror-dest service ID. A SAP that is defined within a mirror destination cannot be used in a mirror source.

```
debug>mirror-source# sap sap-id {[egress] [ingress]}
```

```
*A:ALA-A>debug>mirror-source# sap 2/1/4:100 ingress
egress
```

```
or debug>mirror-source# port 2/2/1.sts12 ingress
```

### 2.6.3.1.3 MAC filter

MAC filters are configured in the **config>filter>mac-filter** context. The **mac-filter** command causes all the packets matching the explicitly defined list of entry IDs to be mirrored to the mirror destination specified by the service-id of the mirror source.

```
debug>mirror-source# mac-filter mac-filter-id entry
entry-id [entry-id ...]
```

```
*A:ALA-2>debug>mirror-source# mac-filter 12 entry 15 20 25
```

### 2.6.3.1.4 IP filter

IP filters are configured in the **config>filter>ip-filter** or **config>filter>ipv6-filter** context. The **ip-filter** command causes all the packets matching the explicitly defined list of entry IDs to be mirrored to the mirror destination specified by the service-id of the mirror source.

Ingress mirrored packets are mirrored to the mirror destination before any ingress packet modifications. Egress mirrored packets are mirrored to the mirror destination after all egress packet modifications.

```
debug>mirror-source# ip-filter ip-filter-id entry entry-
id [entry-id ...]
debug>mirror-source# ipv6-filter ipv6-filter-id entry
```



```
entry-id [entry-id...]
```

```
*A:ALA-A>debug>mirror-source# ip-filter 1 entry 20
```



**Note:** An IP filter cannot be applied to a mirror destination SAP.

### 2.6.3.1.5 Ingress label

The **ingress-label** command is used to mirror ingressing MPLS frames with the specified MPLS labels. The supported MPLS labels are LDP, RSVP, and LDP over RSVP. The ingress label must be at the top of the label stack and can only be mirrored to a single mirror destination. If the same label is defined with multiple mirror destinations, an error is generated and the original mirror destination does not change. The **ingress-label** allows packets matching the ingress label to be duplicated (mirrored) and forwarded to the mirror destination. The ingress label must be active before it can be used as mirror source criteria. If the ingress label is not used in the router, the mirror source removes the ingress label automatically.

```
debug>mirror-source# ingress-label label [label...]
```

```
*A:ALA-A>debug>mirror-source# ingress-label 103000 1048575
```

### 2.6.3.1.6 Subscriber

The subscriber command is used to add hosts of a subscriber to a mirroring service. This command applies to the 7450 ESS and 7750 SR only.

```
debug>mirror-source# subscriber sub-ident-string  
[sap...]
```

```
config>mirror>mirror-source# subscriber sub-ident-string  
[sap...]
```



**Note:** When mirroring an LAC subscriber, family (IPv4 and IPv6) is not applicable. Both IPv4 and IPv6 traffic are mirrored.

## 2.6.4 Common configuration tasks

This section provides a brief overview of the tasks that must be performed to configure both local and remote mirror services and provides the CLI command syntax. Note that local and remote mirror source and mirror destination components must be configured under the same service ID context.

### 2.6.4.1 Configuring a local mirror service

To configure a local mirror service, the source and destinations must be located on the same router. Note that local mirror source and mirror destination components must be configured under the same service ID context.

The **mirror-source** commands are used as traffic selection criteria to identify traffic to be mirrored at the source. Each of these criteria are independent. For example, in the same mirror-source an entire port X could be mirrored at the same time as packets matching a filter entry applied to SAP Y could be mirrored. A filter must be applied to the SAP or interface if only specific packets are to be mirrored. Note that slice-size is not supported by CEM encap-types or IP-mirroring (only applies to the 7750 SR and 7950 XRS).

Use the CLI syntax to configure one or more mirror source parameters:

The **mirror-dest** commands are used to specify where the mirrored traffic is to be sent, the forwarding class, and the size of the packet.

The following output shows an example of a local mirrored service. On ALA-A, mirror service 103 is mirroring traffic matching IP filter 2, entry 1 as well as egress and ingress traffic on port 2/1/24 and sending the mirrored packets to SAP 2/1/25:

```
*A:ALA-A>config>mirror# info
-----
      mirror-dest 103 create
      sap 2/1/25:0 create
egress
      qos 1
      exit
      exit
      no shutdown
      exit
-----
*A:ALA-A>config>mirror#
```

The following output shows debug mirroring information:

```
*A:ALA-A>debug>mirror-source# show debug mirror
debug
  mirror-source 103
  no shutdown
  port 2/1/24 egress ingress
  ip-filter 2 entry 1
  exit
exit
*A:ALA-A>debug>mirror-source# exit
```

The following output shows using **config mirror source** as an alternative:

```
*A:ALA-A>config>mirror# info
  mirror-source 103
  no shutdown
  port 2/1/24 egress ingress
  ip-filter 2 entry 1
  exit
```

The IP filter and entry referenced by the mirror source must exist and must be applied to an object for traffic to be mirrored:

```
*A:ALA-A>config>service>vprn>if# info
-----
      sap 1/1/3:63 create
      ingress
      filter ip 2
      exit
      exit
```

---

## 2.6.4.2 Configuring SDPs for mirrors and LI

This section provides a brief overview of the tasks that must be performed to configure SDPs and provides the CLI commands. For more information about service configuration, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Services Overview Guide*.

Consider the following SDP characteristics:

- Configure GRE, MPLS, MPLS-TP, or L2TPv3 SDPs.
- Each distributed service must have an SDP defined for every remote SR to provide Epipe, VPLS, or mirrored services.
- A distributed service must be bound to an SDP. By default, no SDP is associated with a service. After an SDP is created, services can be associated with that SDP.
- An SDP is not specific to any one service or any type of service. An SDP can have more than one service bound to it.
- When using L2TPv3, MPLS-TP, or LDP IPv6 LSP SDPs in a remote mirroring solution, configure the destination node with **remote-src>spoke-sdp** entries. For all other types of SDPs use **remote-src>far-end** entries.
- To configure an MPLS SDP, LSPs must be configured first and then the LSP-to-SDP association must be explicitly created.

### 2.6.4.2.1 Configuring basic SDPs

#### Prerequisites

To configure a basic SDP, perform the following steps:

#### Procedure

- Step 1.** Select an originating node.
- Step 2.** Create an SDP ID.
- Step 3.** Select an encapsulation type.
- Step 4.** Select the far-end node.

### 2.6.4.2.2 Configuring return path SDPs

#### Prerequisites

To configure the return path SDP, perform the same steps on the far-end router:

#### Procedure

- Step 1.** Select an originating node.
- Step 2.** Create an SDP ID.
- Step 3.** Select an encapsulation type.
- Step 4.** Select the far-end node.

## What to do next

Use the following CLI syntax to create an SDP and select an encapsulation type. If you do not specify a delivery type, the default encapsulation type is GRE.



**Note:** When specifying the far-end IP address, a tunnel is created, the path from Point A to Point B. Use the **show service sdp** command to display the qualifying SDPs.

```
config>service# sdp sdp-id [gre | mpls | l2tpv3 | gre-eth-bridged] create
description description-string
far-end ip-address|ipv6-address
lsp lsp-name [lsp-name]
path-mtu octets
no shutdown
keep-alive
hello-time seconds
hold-down-time seconds
max-drop-count count
message-length octets
no shutdown
timeout timeout
```

On the mirror source router, configure an SDP pointing toward the mirror destination router (or use an existing SDP).

On the mirror destination router, configure an SDP pointing toward the mirror source router (or use an existing SDP).

The following example shows SDP configurations on both the mirror source and mirror destination routers.

```
*A:ALA-A>config>service# info
-----
sdp 1 create
    description "to-10.10.10.104"
    far-end 10.10.10.104
    no shutdown
    exit
-----
*A:ALA-A>config>service#

*A:ALA-B>config>service# info
-----
sdp 4 create
    description "to-10.10.10.103"
    far-end 10.10.10.103
    no shutdown
    exit
-----
*A:ALA-B>config>service#
```

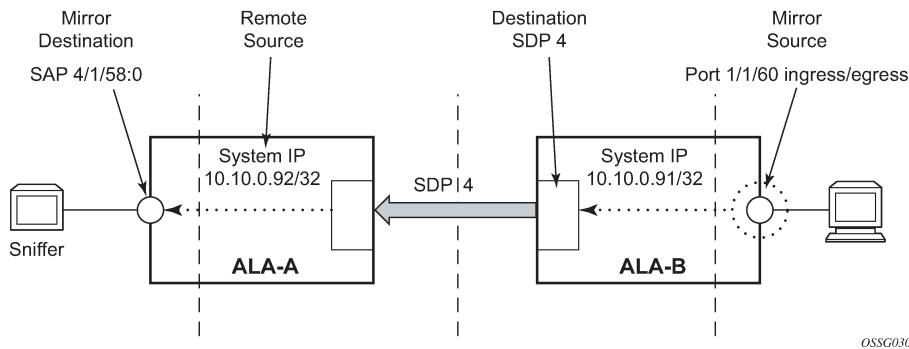
### 2.6.4.3 Configuring a remote mirror service

For remote mirroring, the source and destination are configured on the different routers. Note that mirror source and mirror destination parameters must be configured under the same service ID context.

When using L2TPv3, MPLS-TP or LDP IPv6 LSP spoke SDPs in a remote mirroring solution, configure the destination node with **remote-src>spoke-sdp** entries. For all other types of SDPs use **remote-src>far-end** entries.

**Figure 18: Remote mirrored service tasks** shows the mirror destination, which is on ALA-A, configuration for mirror service 1216. This configuration specifies that the mirrored traffic coming from the mirror source (10.10.0.91) is to be directed to SAP 4/1/58 and states that the service only accepts traffic from far end 10.10.0.92 (ALA-B) with an ingress service label of 5678. When a forwarding class is specified, then all mirrored packets transmitted to the destination SAP or SDP override the default (be) forwarding class. The slice size limits the size of the stream of packet through the router and the core network.

*Figure 18: Remote mirrored service tasks*



The following example shows the CLI output showing the configuration of remote mirrored service 1216. The traffic ingressing and egressing port 1/1/60 on 10.10.0.92 (ALA-B) is mirrored to the destination SAP 1/1/58:0 on ALA-A.

```
*A:ALA-A>config>mirror# info
-----
  mirror-dest 1216 create
    description "Receiving mirror traffic from .91"
    remote-source
      far-end 10.10.0.91 ing-svc-label 5678
    exit
  sap 1/1/58:0 create
    egress
      qos 1
    exit
  exit
  no shutdown
  exit
-----
*A:ALA-A>config>mirror#
```

The following example shows the remote mirror destination configured on ALA-B:

```
*A:ALA-B>config>mirror# info
-----
  mirror-dest 1216 create
    description "Sending mirrored traffic to .92"
    fc h1
    spoke-sdp 4:60 create
    egress
    vc-label 5678
  exit
  no shutdown
  exit
  slice-size 128
  no shutdown
  exit
-----
```

```
-----
*A:ALA-B>config>mirror#
```

The following example shows the mirror source configuration for ALA-B:

```
*A:ALA-B# show debug mirror
debug
  mirror-source 1216
  port 1/1/60 egress ingress
  no shutdown
  exit
exit
*A:ALA-B#
```

The following example is an alternative for mirror source configuration:

```
*A:ALA-B# config>mirror#info
  mirror-source 1216
  port 1/1/60 egress ingress
  no shutdown
  exit
*A:ALA-B#
```

The following example shows the SDP configuration from ALA-A to ALA-B (SDP 2) and the SDP configuration from ALA-B to ALA-A (SDP 4):

```
*A:ALA-A>config>service>sdp# info
-----
  description "GRE-10.10.0.91"
  far-end 10.10.0.01
  no shutdown
-----
*A:ALA-A>config>service>sdp#

*A:ALA-B>config>service>sdp# info
-----
  description "GRE-10.10.20.92"
  far-end 10.10.10.103
  no shutdown
-----
*A:ALA-B>config>service>sdp#
```

#### 2.6.4.4 Configuring Lawful Intercept parameters

The following example shows an LI source configuration and LI log configuration examples:

```
A:ALA-48>config# info
#-----
...
(LI Source Config)
  li-source 1
    sap 1/5/5:1001 egress ingress
    no shutdown
  exit
  li-source 2
    subscriber "test" sla-profile "test" fc l2 ingress egress
    no shutdown
```

```

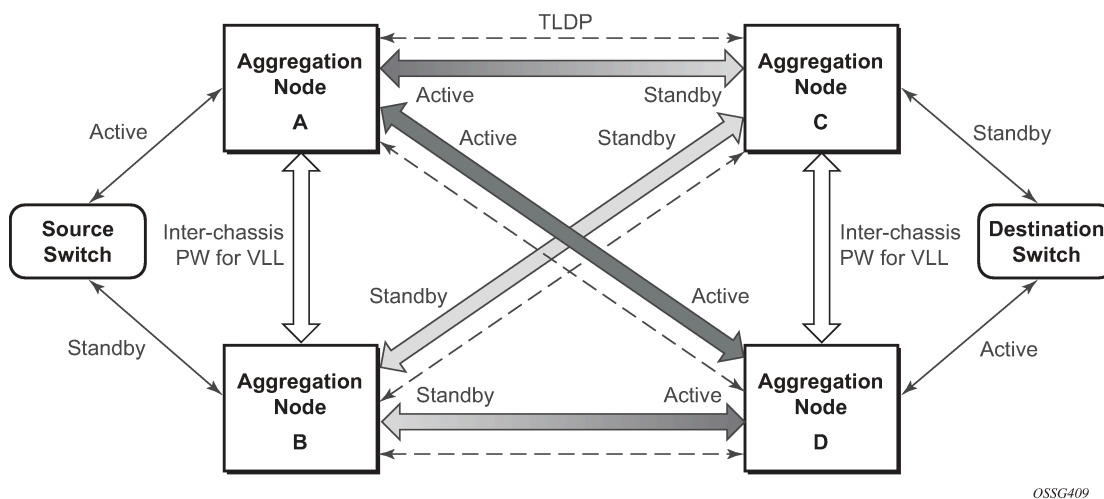
exit
li-source 3
  mac-filter 10 entry 1
  no shutdown
exit
li-source 4
  ip-filter 11 entry 1
  no shutdown
exit
...
(LI Log Config)
log-id 1
  filter 1
  from li
  to session
exit
log-id 11
  from li
  to memory
exit
log-id 12
  from li
  to snmp
exit
...
#-----
A:ALA-48>config#

```

### 2.6.4.5 Pseudowire redundancy for mirror services configuration example

A configuration based on [Figure 19: State engine for redundant service to a redundant mirror service](#) is described in this section.

Figure 19: State engine for redundant service to a redundant mirror service



The mirror traffic needs to be forwarded from configured debug mirror-source together with mirror-dest/ remote-source (ICB or non-ICB) to either SAP endpoint or SDP endpoint.

A SAP endpoint is an endpoint with a SAP and with or without an additional ICB spoke. An SDP endpoint is an endpoint with regular and ICB spokes.

Only one tx-active can be chosen for either SAP endpoint or SDP endpoint. Traffic ingressing into a remote-source ICB has only ingressing traffic while an ICB spoke has only egressing traffic.

The ingressing traffic to a remote-source ICB cannot be forwarded out of another ICB spoke.

The following example shows a high-level summary of a configuration; it is not intended to be syntactically correct:

```

Node A:
config mirror mirror-dest 100
endpoint X
sdp to-C endpoint X
sdp to-D endpoint X
sdp to-B endpoint X icb // connects to B's remote-source IP-A, traffic A->B only
remote-source IP-B icb // connects to B's sdp to-A, traffic B->A only

Node B:
config mirror mirror-dest 100
endpoint X
sdp to-C endpoint X
sdp to-D endpoint X
sdp to-A endpoint X icb // connects to A's remote-source IP-B, traffic B->A only
remote-source IP-A icb // connects to Node A's sdp to-B, traffic A->B only

Node C:
config mirror mirror-dest 100
endpoint X
sap lag-1:0 endpoint X
sdp to-D endpoint X icb // connects to D's remote-source IP-C, traffic C->D only
remote-source IP-A
remote-source IP-B
remote-source IP-D icb // connects to D's sdp to-C, traffic D->C only

Node D:
config mirror mirror-dest 100
endpoint X
sap lag-1:0 endpoint X
sdp to-C endpoint X icb // connects to C's remote-source IP-D, traffic D->C only
remote-source IP-A
remote-source IP-B
remote-source IP-C icb // connects to C's sdp to-D, traffic C->D only

```

## 2.7 Service management tasks

This section describes service management tasks related to service mirroring.

### 2.7.1 Modifying a local mirrored service

Existing mirroring parameters can be modified in the CLI. The changes are applied immediately. The service must be shut down if changes to the SAP are made.

The following example shows the commands to modify parameters for a basic local mirroring service:

```

config>mirror# mirror-dest 103
config>mirror>mirror-dest# shutdown
config>mirror>mirror-dest# no sap
config>mirror>mirror-dest# sap 3/1/5:0 create

```



```
config>mirror>mirror-dest>sap$ exit
config>mirror>mirror-dest# fc be
config>mirror>mirror-dest# slice-size 128
config>mirror>mirror-dest# no shutdown
```

```
debug# mirror-dest 103
debug>mirror-source# no port 2/1/24 ingress egress
debug>mirror-source# port 3/1/7 ingress egress
```

The following output shows the local mirrored service modifications:

```
*A:ALA-A>config>mirror# info
-----
mirror-dest 103 create
          no shutdown
          fc be
          remote-source
          exit
          sap 3/1/5:0 create
egress
          qos 1
          exit
          exit
          slice-size 128
          exit

*A:ALA-A>debug>mirror-source# show debug mirror
debug
  mirror-source 103
  no shutdown
  port 3/1/7 egress ingress
exit
*A:ALA-A>debug>mirror-source#
```

## 2.7.2 Deleting a local mirrored service

Existing mirroring parameters can be deleted in the CLI. A **shutdown** must be issued on a service level to delete the service. It is not necessary to shut down or remove SAP or port references to delete a local mirrored service.

The following example shows the commands to delete a local mirrored service.

```
ALA-A>config>mirror# mirror-dest 103
config>mirror>mirror-dest# shutdown
config>mirror>mirror-dest# exit
config>mirror# no mirror-dest 103
config>mirror# exit
```

## 2.7.3 Modifying a remote mirrored service

Existing mirroring parameters can be modified in the CLI. The changes are applied immediately. The service must be shut down if changes to the SAP are made.

In the following example, the mirror destination is changed from 10.10.10.2 (ALA-B) to 10.10.10.3 (SR3). Note that the mirror-dest service ID on ALA-B must be shut down first before it can be deleted.

The following example shows the commands to modify parameters for a remote mirrored service:

```
*A:ALA-A>config>mirror# mirror-dest 104
config>mirror>mirror-dest# remote-source
config>mirror>mirror-dest>remote-source# no far-end
10.10.10.2
remote-source# far-end 10.10.10.3 ing-svc-label 3500

*A:ALA-B>config>mirror# mirror-dest 104
config>mirror>mirror-dest# shutdown
config>mirror>mirror-dest# exit
config>mirror# no mirror-dest 104

SR3>config>mirror# mirror-dest 104 create
config>mirror>mirror-dest# spoke-sdp 4:60 egress vc-
label 3500
config>mirror>mirror-dest# no shutdown
config>mirror>mirror-dest# exit all

SR3># debug
debug# mirror-source 104
debug>mirror-source# port 551/1/2 ingress egress
debug>mirror-source# no shutdown
```

```
*A:ALA-A>config>mirror# info
-----
mirror-dest 104 create
    remote-source
        far-end 10.10.10.3 ing-svc-label 3500
    exit
    sap 2/1/15:0 create
egress
    qos 1
    exit
    exit
    no shutdown
exit

A:SR3>config>mirror# info
-----
    mirror-dest 104 create
        spoke-sdp 4:60 egress vc-label 3500
        no shutdown
    exit
-----

A:SR3>config>mirror#

A:SR3# show debug mirror
debug
    mirror-source 104
        no shutdown
        port 5/1/2 egress ingress
exit
    exit
A:SR3#
```

## 2.7.4 Deleting a remote mirrored service

Existing mirroring parameters can be deleted in the CLI. A shut down must be issued on a service level to delete the service. It is not necessary to shut down or remove SAP, SDP, or far-end references to delete a remote mirrored service.

Mirror destinations must be shut down first before they can be deleted.

```
*A:ALA-A>config>mirror# mirror-dest 105
config>mirror>mirror-dest# shutdown
config>mirror>mirror-dest# exit
config>mirror# no mirror-dest 105
config>mirror# exit
```

```
*A:ALA-B>config>mirror# mirror-dest 105
config>mirror>mirror-dest# shutdown
config>mirror>mirror-dest# exit
config>mirror# no mirror-dest 105
config>mirror# exit
```

In the example, the mirror-destination service ID 105 was removed from the configuration on ALA-A and ALA-B, therefore, it does not appear in the info command output.

```
*A:ALA-A>config>mirror# info
```

```
-----
-----
*A:ALA-A>config>mirror# exit
```

```
*A:ALA-B>config>mirror# info
```

```
-----
-----
*A:ALA-B>config>mirror# exit
```

Because the mirror destination was removed from the configuration on ALA-B, the port information was automatically removed from the debug mirror-source configuration.

```
*A:ALA-B# show debug mirror
debug
exit
*A:ALA-B#
```

## 3 OAM fault and performance tools and protocols

### 3.1 OAM overview

Delivery of services requires a number of operations occur properly and at different levels in the service delivery model. For example, operations such as the association of packets to a service, VC-labels to a service and each service to a service tunnel must be performed properly in the forwarding plane for the service to function properly. To verify that a service is operational, a set of in-band, packet-based Operation, Administration, and Maintenance (OAM) tools is required, with the ability to test each of the individual packet operations.

For in-band testing, the OAM packets closely resemble customer packets to effectively test the customer's forwarding path, but they are distinguishable from customer packets so they are kept within the service provider's network and not forwarded to the customer.

The suite of OAM diagnostics supplement the basic IP ping and traceroute operations with diagnostics specialized for the different levels in the service delivery model. There are diagnostics for MPLS LSPs, SDPs, services and VPLS MACs within a service.

#### 3.1.1 LSP diagnostics: LSP ping and LSP trace

The router LSP diagnostics are implementations of LSP ping and LSP trace based on RFC 8029, *Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures*. LSP ping provides a mechanism to detect data plane failures in MPLS LSPs. LSP ping and LSP trace are modeled after the ICMP echo request or reply used by ping and trace to detect and localize faults in IP networks.

For a specific LDP FEC, RSVP P2P LSP, or BGP IPv4 or IPv6 label route, LSP ping verifies whether the packet reaches the egress label edge router (LER), while in LSP trace mode, the packet is sent to the control plane of each transit label switched router (LSR) which performs various checks to see if it is actually a transit LSR for the path.

The downstream mapping TLV is used in lsp-ping and lsp-trace to provide a mechanism for the sender and responder nodes to exchange and validate interface and label stack information for each downstream hop in the path of an LDP FEC or an RSVP LSP.

Two downstream mapping TLVs are supported. The original Downstream Mapping (DSMAP) TLV defined in RFC 4379, *Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures*, (obsoleted by RFC 8029) and the new Downstream Detailed Mapping (DDMAP) TLV defined in RFC 6424, *Mechanism for Performing Label Switched Path Ping (LSP Ping) over MPLS Tunnels*, and RFC 8029.

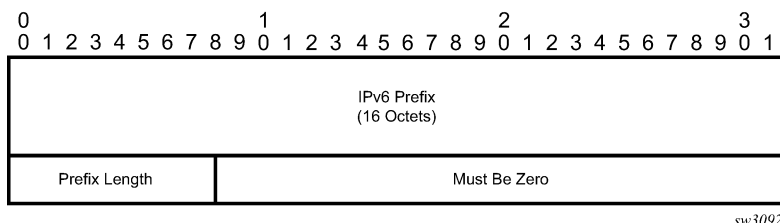
When the responder node has multiple equal cost next-hops for an LDP FEC prefix, the downstream mapping TLV can further be used to exercise a specific path of the ECMP set using the path-destination option. The behavior in this case is described in the ECMP sub-section below.

### 3.1.2 LSP ping and LSP trace for an LSP using a BGP IPv4 or IPv6 label route

This feature adds support of the Target FEC Stack TLV of type BGP Labeled IPv4 /32 Prefix as defined in RFC 8029.

The new TLV is structured as shown in [Figure 20: Target FEC stack TLV for a BGP labeled IPv4 and IPv6 prefixes](#).

Figure 20: Target FEC stack TLV for a BGP labeled IPv4 and IPv6 prefixes



The user issues a LSP ping using the existing CLI command and specifying a new type of prefix:

```
oam lsp-ping bgp-label prefix ip-prefix/mask [src-ip-address ip-address] [fc fc-name [profile {in | out}]]  
[size octets] [ttl label-ttl] [send-count send-count] [timeout timeout] [interval interval] [path-destination  
ip-address [interface if-name | next-hop ip-address]] [detail]
```

This feature supports BGP label IPv4 prefixes with a prefix length of 32 bits only and supports IPv6 prefixes with a prefix length of 128 bits only.

The **path-destination** option is used to exercise specific ECMP paths in the network when the LSR performs hashing on the MPLS packet.

Similarly, the user issues a LSP trace using the following command:

```
oam lsp-trace bgp-label prefix ip-prefix/mask [src-ip-address ip-address] [fc fc-name [profile {in | out}]]  
[max-fail no-response-count] [probe-count probes-per-hop] [size octets] [min-ttl min-label-ttl] [max-ttl  
max-label-ttl] [timeout timeout] [interval interval] [path-destination ip-address [interface if-name | next-  
hop ip-address]] [detail]
```

The following is the process to send and respond to an LSP ping or LSP trace packet. This process is valid when the downstream mapping is set to the DSMAP TLV. The detailed procedures with the DDMAP TLV are presented in [Using DDMAP TLV in LSP stitching and LSP hierarchy](#).

- The next-hop of a BGP label route for a IPv4 /32 or an IPv6 /128 can be resolved to either an IPv4 transport tunnel or to an IPv6 transport tunnel. Thus, the sender node encapsulates the packet of the echo request message with a label stack which consists of the transport label stack as the outer labels and the BGP label as the inner label.

If the packet expires on a node which acts as an LSR for the outer transport LSP, and does not have context for the BGP label prefix, the outer label in the stack is validated and if the validation is successful it replies as in the case when it receives an echo request message for an LDP FEC which is stitched to a BGP IPv4 label route. In other words, it replies with return code 8 Label switched at stack-depth <RSC>.

- An LSR node which is the next-hop for the BGP label prefix as well as the LER node which originated the BGP label prefix have full context for the BGP IPv4 or IPv6 target FEC stack and can therefore perform full validation of it.
- If a BGP IPv4 label route is stitched to an LDP FEC, the egress LER for the resulting LDP FEC does not have context for the BGP IPv4 target FEC stack in the echo request message and replies with return

code 4 Replying router has no mapping for the FEC at stack- depth <RSC>. This is the same behavior as that of an LDP FEC which is stitched to a BGP IPv4 label route when the echo request message reaches the egress LER for the BGP prefix.

Note that only BGP label IPv4 /32 prefixes and BGP IPv6 /128 prefixes are supported because only these are usable as tunnels on the Nokia router platforms. The BGP IPv4 or IPv6 label prefix is also supported with the prefix SID attribute if BGP segment routing is enabled on the routers participating in the path of the tunnel.

The responder node must have an IPv4 address to use as the source address of the IPv4 echo reply packet. SR OS uses the system interface IPv4 address. When an IPv4 BGP label route resolves to an IPv6 next-hop and uses an IPv6 transport tunnel, any LSR or LER node which responds to an lsp-ping or lsp-trace message must have an IPv4 address assigned to the system interface or the reply is not sent. In the latter case, the lsp-ping or lsp-trace probe times out at the sender node.

Similarly, the responder node must have an IPv6 address assigned to the system interface so that it gets used in the IPv6 echo reply packet in the case of a BGP-LU IPv6 label route when resolved to an IPv4 or an IPv4-mapped IPv6 next-hop which itself is resolved to an IPv4 transport tunnel.

### 3.1.3 ECMP considerations

When the responder node has multiple equal cost next-hops for an LDP FEC or a BGP label prefix, it replies in the Downstream Mapping TLV with the downstream information of the outgoing interface which is part of the ECMP next-hop set for the prefix.

Note that when BGP label route is resolved to an LDP FEC (of the BGP next-hop of the BGP label route), ECMP can exist at both the BGP and LDP levels. The following selection of next hop is performed in this case:

- For each BGP ECMP next-hop of the label route, a single LDP next-hop is selected even if multiple LDP ECMP next-hops exist. Thus, the number of ECMP next-hops for the BGP label route is equal to the number of BGP next-hops.
- ECMP for a BGP label route is only supported at PE router (BGP label push operation) and not at ABR/ASBR (BGP label swap operation). Thus at an LSR, a BGP label route is resolved to a single BGP next-hop which itself is resolved to a single LDP next-hop.
- LSP trace returns one downstream mapping TLV for each next-hop of the BGP label route. Furthermore, it returns exactly the LDP next-hop the data path programmed for each BGP next-hop.

The following description of the behavior of LSP ping and LSP trace makes a reference to a FEC in a generic way and which can represent an LDP FEC or a BGP label route. In addition, the reference to a downstream mapping TLV means either the DSMAP TLV or the DDMAP TLV.

- If the user initiates an lsp-trace of the FEC without the **path-destination** option specified, then the sender node does not include multi-path information in the Downstream Mapping TLV in the echo request message (multipath type=0). In this case, the responder node replies with a Downstream Mapping TLV for each outgoing interface which is part of the ECMP next-hop set for the FEC. Note that the sender node selects the first Downstream Mapping TLV only for the subsequent echo request message with incrementing TTL.
- If the user initiates an lsp-ping of the FEC with the **path-destination** option specified, then the sender node does not include the Downstream Mapping TLV. However, the user can use the **interface** option, part of the same **path-destination** option, to direct the echo request message at the sender node to be sent out a specific outgoing interface which is part of an ECMP path set for the FEC.

- If the user initiates an `lsp-trace` of the FEC with the **path-destination** option specified but configured not to include a downstream mapping TLV in the MPLS echo request message using the CLI command **downstream-map-tlv {none}**, then the sender node does not include the Downstream Mapping TLV. However, the user can use the **interface** option, part of the same **path-destination** option, to direct the echo request message at the sender node to be sent out a specific outgoing interface which is part of an ECMP path set for the FEC.
- If the user initiates an `lsp-trace` of the FEC with the **path-destination** option specified, then the sender node includes the multipath information in the Downstream Mapping TLV in the echo request message (multipath type=8). The **path-destination** option allows the user to exercise a specific path of a FEC in the presence of ECMP. This is performed by having the user enter a specific address from the 127/8 range which is then inserted in the multipath type 8 information field of the Downstream Mapping TLV. The CPM code at each LSR in the path of the target FEC runs the same hash routine as the data path and replies in the Downstream Mapping TLV with the specific outgoing interface the packet would have been forwarded to if it did not expire at this node and if DEST IP field in the packet's header was set to the 127/8 address value inserted in the multipath type 8 information. This hash is based on:
  - the {incoming port, system interface address, label-stack} when the **lsp-load-balancing** option of the incoming interface is configured to **lbl-only**. In this case the 127/8 prefix address entered in the **path-destination** option is not used to select the outgoing interface. All packets received with the same label stack maps to a single and same outgoing interface.
  - the {incoming port, system interface address, label-stack, SRC/DEST IP fields of the packet} when the **lsp-load-balancing** option of the incoming interface is configured to **lbl-ip**. The SRC IP field corresponds to the value entered by the user in the **src-ip-address** option (default system IP interface address). The DEST IP field corresponds to the 127/8 prefix address entered in the **path-destination** option. In this case, the CPM code maps the packet, as well as any packet in a sub-range of the entire 127/8 range, to one of the possible outgoing interface of the FEC.
  - the {SRC/DEST IP fields of the packet} when the **lsp-load-balancing** option of the incoming interface is configured to **ip-only**. The SRC IP field corresponds to the value entered by the user in the **src-ip-address** option (default system IP interface address). The DEST IP field corresponds to the 127/8 prefix address entered in the **path-destination** option. In this case, the CPM code maps the packet, as well as any packet in a sub-range of the entire 127/8 range, to one of the possible outgoing interface of the FEC.

In all above cases, the user can use the **interface** option, part of the same **path-destination** option, to direct the echo request message at the sender node to be sent out a specific outgoing interface which is part of an ECMP path set for the FEC.

Note that if the user enabled the **system-ip-load-balancing hash** option (`config>system>system-ip-load-balancing`), then the LSR hashing is modified by applying the system IP interface, with differing bit-manipulation, to the hash of packets of all three options (**lbl-only**, **lbl-ip**, **ip-only**). This system level option enhances the LSR packet distribution such that the probability of the same flow selecting the same ECMP interface index or LAG link index at two consecutive LSR nodes is minimized.

- The **ldp-treetrace** tool always uses the multipath type=8 and inserts a range of 127/8 addresses instead of a single address in order multiple ECMP paths of an LDP FEC. As such, it behaves the same way as the **lsp-trace** with the **path-destination** option enabled described above.
- Note that the **path-destination** option can also be used to exercise a specific ECMP path of an LDP FEC, which is tunneled over a RSVP LSP or of an LDP FEC stitched to a BGP FEC in the presence of BGP ECMP paths. The user must however enable the use of the new DDMAP TLV either globally (`config>test-oam>mpls-echo-request-downstream-map ddmmap`) or within the specific **ldp-treetrace** or **lsp-trace test** (`downstream-map-tlv ddmmap option`).

### 3.1.4 LSP ping and LSP trace over unnumbered IP interface

Lsp-ping and p2mp-lsp-ping operate over a network using unnumbered links without any changes. Lsp-trace, p2mp-lsp-trace and ldp-tree-trace are modified such that the unnumbered interface is properly encoded in the downstream mapping (DSMAP/DDMAP) TLV.

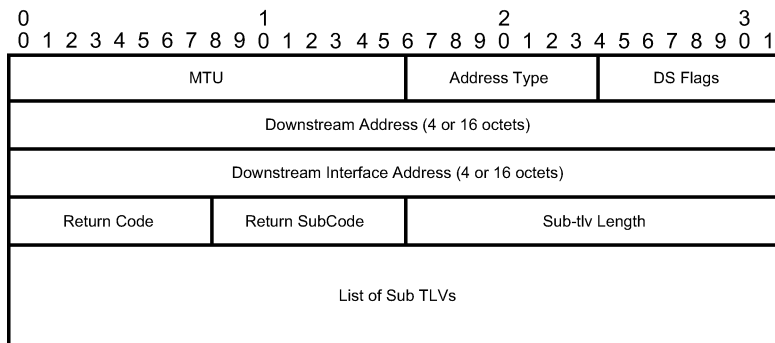
In a RSVP P2P or P2MP LSP, the upstream LSR encodes the downstream router-id in the "Downstream IP Address" field and the local unnumbered interface index value in the "Downstream Interface Address" field of the DSMAP/DDMAP TLV as per RFC 8029. Both values are taken from the TE database.

In a LDP unicast FEC or mLDP P2MP FEC, the interface index assigned by the peer LSR is not readily available to the LDP control plane. In this case, the alternative method described in RFC 8029 is used. The upstream LSR sets the Address Type to IPv4 Unnumbered, the Downstream IP Address to a value of 127.0.0.1, and the interface index is set to 0. If an LSR receives an echo-request packet with this encoding in the DSMAP/DDMAP TLV, it bypasses interface verification but continues with label validation.

### 3.1.5 DDMAP TLV

The DDMAP TLV provides the same features as the existing DSMAP TLV, plus the enhancement to trace the details of LSP stitching and LSP hierarchy. The latter is achieved using a new sub-TLV of the DDMAP TLV called the FEC stack change sub-TLV. [Figure 21: DDMAP TLV](#) shows the structures of these two objects as defined in RFC 6424.

Figure 21: DDMAP TLV

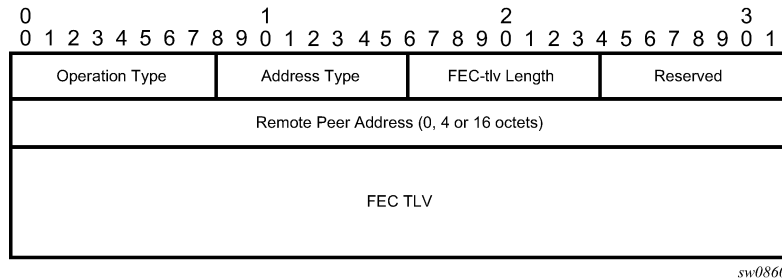


sw0862

The DDMAP TLV format is derived from the DSMAP TLV format. The key change is that variable length and optional fields have been converted into sub-TLVs. The fields have the same use and meaning as in RFC 8029 as shown in [Figure 22: FEC stack change sub-TLV](#).



Figure 22: FEC stack change sub-TLV



The operation type specifies the action associated with the FEC stack change. The following operation types are defined.

Type #	Operation
1	Push
2	Pop

More details on the processing of the fields of the FEC stack change sub-TLV are provided later in this section.

The user can configure which downstream mapping TLV to use globally on a system by using the following command:

**configure test-oam mpls-echo-request-downstream-map {dsmap | ddmmap}**

This command specifies which format of the downstream mapping TLV to use in all LSP trace packets and LDP tree trace packets originated on this node. The Downstream Mapping (DSMAP) TLV is the original format in RFC 4379 (obsoleted by RFC 8029) and is the default value. The Downstream Detailed Mapping (DDMAP) TLV is the new enhanced format specified in RFC 6424 and RFC 8029.

This command applies to LSP trace of an RSVP P2P LSP, a MPLS-TP LSP, a BGP label route, or LDP unicast FEC, and to LDP tree trace of a unicast LDP FEC. It does not apply to LSP trace of an RSVP P2MP LSP which always uses the DDMAP TLV.

The global Downstream Mapping TLV setting impacts the behavior of both OAM LSP trace packets and SAA test packets of type **lsp-trace** and is used by the sender node when one of the following events occurs:

- An SAA test of type **lsp-trace** is created (not modified) and no value is specified for the per-test **downstream-map-tlv {dsmap | ddmmap | none}** option. In this case the SAA test **downstream-map-tlv** value defaults to the global **mpls-echo-request-downstream-map** value.
- An OAM test of type **lsp-trace** test is executed and no value is specified for the per-test **downstream-map-tlv {dsmap | ddmmap | none}** option. In this case, the OAM test **downstream-map-tlv** value defaults to the global **mpls-echo-request-downstream-map** value.

A consequence of the rules above is that a change to the value of **mpls-echo-request-downstream-map** option does not affect the value inserted in the downstream mapping TLV of existing tests.

The following are the details of the processing of the new DDMAP TLV:

- When either the DSMAP TLV or the DDMAP TLV is received in an echo request message, the responder node includes the same type of TLV in the echo reply message with the correct downstream interface information and label stack information.

- If an echo request message without a Downstream Mapping TLV (DSMAP or DDMAP) expires at a node which is not the egress for the target FEC stack, the responder node always includes the DSMAP TLV in the echo reply message. This can occur in the following cases:
  - The user issues a LSP trace from a sender node with a **min-ttl** value higher than 1 and a **max-ttl** value lower than the number of hops to reach the egress of the target FEC stack. This is the sender node behavior when the global configuration or the per-test setting of the Downstream Mapping TLV is set to DSMAP.
  - The user issues a LSP ping from a sender node with a **tll** value lower than the number of hops to reach the egress of the target FEC stack. This is the sender node behavior when the global configuration of the Downstream Mapping TLV is set to DSMAP.
  - The behavior in (a) is changed when the global configuration or the per-test setting of the Downstream Mapping TLV is set to DDMAP. The sender node includes in this case the DDMAP TLV with the Downstream IP address field set to the all-routers multicast address as per Section 3.4 of RFC 8029. The responder node then bypasses the interface and label stack validation and replies with a DDMAP TLV with the correct downstream information for the target FEC stack.
- A sender node never includes the DSMAP or DDMAP TLV in an lsp-ping message.

### 3.1.6 Using DDMAP TLV in LSP stitching and LSP hierarchy

In addition to performing the same features as the DSMAP TLV, the DDMAP TLV addresses the following scenarios:

- Full validation of an LDP IPv4 FEC stitched to a BGP IPv4 label route. In this case, the LSP trace message is inserted from the LDP LSP segment or from the stitching point.
- Full validation of a BGP IPv4 label route stitched to an LDP IPv4 FEC. The LSP trace message is inserted from the BGP LSP segment or from the stitching point.
- Full validation of an LDP IPv4 FEC which is stitched to a BGP IPv4 label route and stitched back into an LDP IPv4 FEC. In this case, the LSP trace message is inserted from the LDP segments or from the stitching points.
- Full validation of a LDP IPv4 FEC stitched to a SR-ISIS or SR-OSPF IPv4 tunnel.
- Full validation of an SR-ISIS or SR-OSPF IPv4 tunnel stitched to an LDP IPv4 FEC.
- Full validation of an LDP FEC tunneled over an RSVP LSP or an SR-TE LSP using LSP trace.
- Full validation of a BGP IPv4 label route or of a BGP IPv6 label route (with an IPv4 or an IPv4-mapped IPv6 next-hop) tunneled over an RSVP LSP, an LDP IPv4 FEC, an SR-ISIS IPv4 tunnel, a SR-OSPF IPv4 tunnel, an SR-TE IPv4 LSP, or an IPv4 SR policy.
- Full validation of a BGP IPv4 label route (with an IPv6 next-hop) or a BGP IPv6 label route tunneled over an LDP IPv6 FEC, an SR-ISIS IPv6 tunnel, an SR-OSPF3 IPv6 tunnel, an SR-TE IPv6 LSP, or an IPv6 SR policy.
- Full validation of a BGP IPv6 label route (with an IPv4 or an IPv4-mapped IPv6 next-hop) recursively resolved to a BGP IPv4 label route which itself is tunneled over an LDP IPv4 FEC, an SR-ISIS IPv4 tunnel, an SR-OSPF IPv4 tunnel, an RSVP-TE LSP, an SR-TE IPv4 LSP, or an IPv4 SR policy.

To properly check a target FEC which is stitched to another FEC (stitching FEC) of the same or a different type, or which is tunneled over another FEC (tunneling FEC), it is necessary for the responding nodes to provide details about the FEC manipulation back to the sender node. This is achieved via the use of the new FEC stack change sub-TLV in the Downstream Detailed Mapping TLV (DDMAP) defined in RFC 6424.

When the user configures the use of the DDMAP TLV on a trace for an LSP that does not undergo stitching or tunneling operation in the network, the procedures at the sender and responder nodes are the same as in the case of the existing DSMAP TLV.

This feature however introduces changes to the target FEC stack validation procedures at the sender and responder nodes in the case of LSP stitching and LSP hierarchy. These changes pertain to the processing of the new FEC stack change sub-TLV in the new DDMAP TLV and the new return code 15 Label switched with FEC change. The following is a description of the main changes which are a superset of the rules described in Section 4 of RFC 6424 to allow greater scope of interoperability with other vendor implementations.

### 3.1.6.1 Responder node procedures

This section describes responder-node behaviors.

- As a responder node, the router always inserts a global return code of either:
  - 3 Replying router is an egress for the FEC at stack-depth <RSC>
  - 14 See DDMAP TLV for Return Code and Return Subcode.
- When the responder node inserts a global return code of 3, it does not include a DDMAP TLV.
- When the responder node includes the DDMAP TLV, it inserts a global return code, 14 See DDMAP TLV for Return Code and Return Subcode and:
  - On a success response, include a return code of 15 in the DDMAP TLV for each downstream which has a FEC stack change TLV.
  - On a success response, include a return code 8 Label switched at stack-depth <RSC> in the DDMAP TLV for each downstream if no FEC stack change sub-TLV is present.
  - On a failure response, include an appropriate error return code in the DDMAP TLV for each downstream.
- A tunneling node indicates that it is pushing a FEC (the tunneling FEC) on top of the Target FEC Stack TLV by including a FEC stack change sub-TLV in the DDMAP TLV with a FEC operation type value of PUSH. It also includes a return code 15 Label switched with FEC change. The downstream interface address and downstream IP address fields of the DDMAP TLV are populated for the pushed FEC. The remote peer address field in the FEC stack change sub-TLV is populated with the address of the control plane peer for the pushed FEC. The Label stack sub-TLV provides the full label stack over the downstream interface.
- A node that is stitching a FEC indicates that it is performing a POP operation for the stitched FEC followed by a PUSH operation for the stitching FEC and potentially one PUSH operation for the transport tunnel FEC. It includes two or more FEC stack change sub-TLVs in the DDMAP TLV in the echo reply message. It also includes a return code 15 Label switched with FEC change. The downstream interface address and downstream address fields of the DDMAP TLV are populated for the stitching FEC. The remote peer address field in the FEC stack change sub-TLV of type POP is populated with a null value (0.0.0.0). The remote peer address field in the FEC stack change sub-TLV of type PUSH is populated with the address of the control plane peer for the tunneling FEC. The Label stack sub-TLV provides the full label stack over the downstream interface.
- If the responder node is the egress for one or more FECs in the target FEC Stack, then it must reply with no DDMAP TLV and with a return code 3 Replying router is an egress for the FEC at stack-depth <RSC>. RSC must be set to the depth of the topmost FEC. This operation is iterative in a sense that at the receipt of the echo reply message the sender node pops the topmost FEC from the target stack

FEC TLV and resend the echo request message with the same TTL value as described in (5) below. The responder node performs exactly the same operation as described in this step until all FECs are popped or until the topmost FEC in the Target FEC Stack TLV matches the tunneled or stitched FEC. In the latter case, processing of the Target FEC Stack TLV follows again steps (1) or (2).

### 3.1.6.2 Sender node procedures

This section describes sender-node behaviors.

- If the echo reply message contains the return code 14 See DDMAP TLV for Return Code and Return Subcode and the DDMAP TLV has a return code 15 Label switched with FEC change, the sender node adjusts the target FEC Stack TLV in the echo request message for the next value of the TTL to reflect the operation on the current target FEC stack as indicated in the FEC stack change sub-TLV received in the DDMAP TLV of the last echo reply message. In other words, one FEC is popped at most and one or more FECs are pushed as indicated.
- If the echo reply message contains the return code 3 Replying router is an egress for the FEC at stack-depth <RSC>, then:
  - If the value for the label stack depth specified in the Return Sub-Code (RSC) field is the same as the depth of current target FEC Stack TLV, then the sender node considers the trace operation complete and terminates it. A responder node causes this case to occur as per step (6) of the responder node procedures.
  - If the value for the label stack depth specified in the Return Sub-Code (RSC) field is different from the depth of the current target FEC Stack TLV, the sender node must continue the LSP trace with the same TTL value after adjusting the Target FEC Stack TLV by removing the top FEC. Note that this step continues iteratively until the value for the label stack depth specified in the Return Sub-Code (RSC) field is the same as the depth of current target FEC Stack TLV and in which case step (a) is performed. A responder node causes this case to occur as per step (6) of the responder node procedures.
  - If a DDMAP TLV with or without a FEC stack change sub-TLV is included, then the sender node must ignore it and processing is performed as per steps (a) or (b) above. A responder node does not cause this case to occur but a third party implementation may.
- As a sender node, the router can accept an echo-reply message with the global return code of either 14 (with DDMAP TLV return code of 15 or 8), or 15 and process properly the FEC stack change TLV as per step (1) of the sender node procedures.
- If an LSP ping is performed directly to the egress LER of the stitched FEC, there is no DDMAP TLV included in the echo request message and the responder node, which is the egress node, replies with return code 4 Replying router has no mapping for the FEC at stack- depth <RSC>. This case cannot be resolved with this feature.

### 3.1.7 MPLS OAM support in Segment Routing

MPLS OAM supports Segment Routing extensions to **lsp-ping** and **lsp-trace** as specified in *draft-ietf-mpls-spring-lsp-ping*.

Segment Routing (SR) performs both shortest path and source-based routing. When the data plane uses MPLS encapsulation, MPLS OAM tools such as **lsp-ping** and **lsp-trace** can be used to check connectivity and trace the path to any mid-point or endpoint of an SR-ISIS, a SR-OSPF shortest path tunnel, or an SR-TE LSP.

The CLI options for **lsp-ping** and **lsp-trace** are under OAM and SAA for the following types of Segment Routing tunnels:

- SR-ISIS and SR-OSPF node SID tunnels
- SR-TE LSP

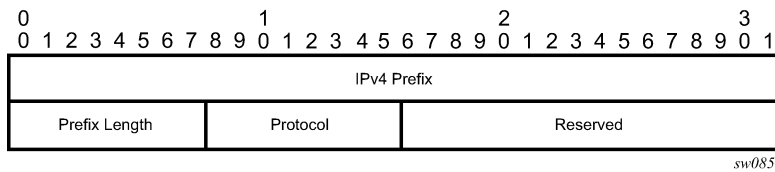
### 3.1.7.1 SR extensions for LSP ping and LSP trace

This section describes how MPLS OAM models the SR tunnel types.

An SR shortest path tunnel, SR-ISIS, or SR-OSPF tunnel, uses a single FEC element in the Target FEC Stack TLV. The FEC corresponds to the prefix of the node SID in a specific IGP instance.

Figure 23: IPv4 IGP-prefix segment ID illustrates the format for the IPv4 IGP-prefix segment ID:

Figure 23: IPv4 IGP-prefix segment ID

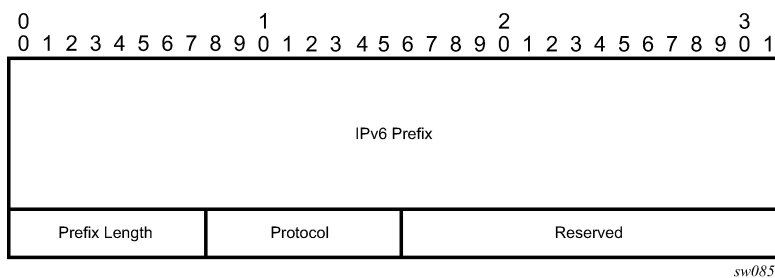


In this format, the fields are as follows:

- **IPv4 Prefix**  
This field carries the IPv4 prefix to which the segment ID is assigned. For anycast segment ID, this field carries the IPv4 anycast address. If the prefix is shorter than 32 bits, trailing bits must be set to zero.
- **Prefix Length**  
The Prefix Length field is one octet. It gives the length of the prefix in bits (values can be 1 to 32).
- **Protocol**  
This field is set to 1 if the IGP protocol is OSPF and is set to 2 if the IGP protocol is IS-IS.

Figure 24: IPv6 IGP prefix segment ID illustrates the format for the IPv6 IGP prefix segment ID:

Figure 24: IPv6 IGP prefix segment ID



In this format, the fields are as follows:

- **IPv6 Prefix**  
This field carries the IPv6 prefix to which the segment ID is assigned. For anycast segment ID, this field carries the IPv4 anycast address. If the prefix is shorter than 128 bits, trailing bits must be set to zero.

- **Prefix Length**

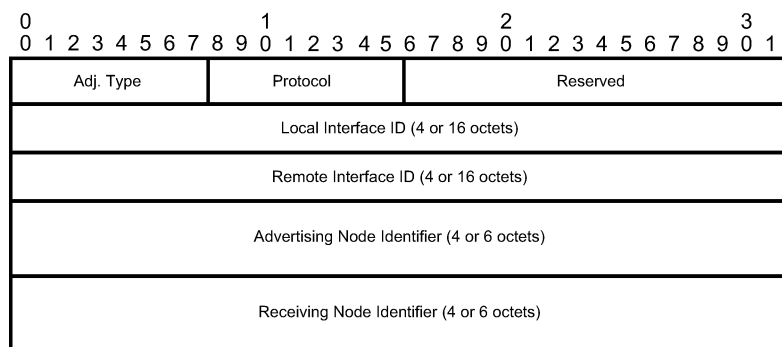
The Prefix Length field is one octet, it gives the length of the prefix in bits (values can be 1 to 128).

- **Protocol**

This field is set to 1 if the IGP protocol is OSPF and is set to 2 if the IGP protocol is IS-IS.

An SR-TE LSP, as a hierarchical LSP, uses the Target FEC Stack TLV, which contains a FEC element for each node SID and for each adjacency SID in the path of the SR-TE LSP. Because the SR-TE LSP does not instantiate state in the LSR other than the ingress LSR, MPLS OAM is just testing a hierarchy of node SID and adjacency SID segments toward the destination of the SR-TE LSP. The format of the node-SID is as illustrated above. [Figure 25: IGP-Adjacency segment ID](#) illustrates the format for the IGP-Adjacency segment ID is as follows:

Figure 25: IGP-Adjacency segment ID



sw0859

In this format, the fields are as follows:

- **Adj. Type (Adjacency Type)**

This field is set to 1 when the adjacency segment is parallel adjacency as defined in section 3.5.1 of *I-D.ietf-spring-segment-routing*. This field is set to 4 when the adjacency segment is IPv4-based and is not a parallel adjacency. This field is set to 6 when the adjacency segment is IPv6-based and is not a parallel adjacency.

- **Protocol**

This field is set to 1 if the IGP protocol is OSPF and is set to 2 if the IGP protocol is IS-IS.

- **Local Interface ID**

This field is an identifier that is assigned by local LSR for a link on which the adjacency segment ID is bound. This field is set to local link address (IPv4 or IPv6). If unnumbered, the 32-bit link identifier defined in RFC 4203 and RFC 5307 is used. If the adjacency segment ID represents parallel adjacencies, as described in section 3.5.1 of *I-D.ietf-spring-segment-routing*, then this field must be set to zero.

- **Remote Interface ID**

This field is an identifier that is assigned by remote LSR for a link on which adjacency segment ID is bound. This field is set to the remote (downstream neighbor) link address (IPv4 or IPv6). If unnumbered, the 32-bit link identifier defined in RFC 4203 and RFC 5307 is used. If the adjacency segment ID represents parallel adjacencies, as described in section 3.5.1 of *I-D.ietf-spring-segment-routing*. This field must be set to zero.

- **Advertising Node Identifier**

This field specifies the advertising node identifier. When the Protocol field is set to 1, then the 32 rightmost bits represent the OSPF router ID. If the Protocol field is set to 2, this field carries the 48-bit IS-IS system ID.

- **Receiving Node Identifier**

This field specifies the downstream node identifier. When the Protocol field is set to 1, then the 32 rightmost bits represent OSPF router ID. If the Protocol field is set to 2, this field carries the 48-bit IS-IS system ID.

Both **Isp-ping** and **Isp-trace** apply to the following contexts:

- SR-ISIS or SR-OSPF shortest path IPv4 tunnel
- SR-ISIS or SR-OSPF3 (OSPFv3 instance ID 0-31) shortest path IPv6 tunnel
- IS-IS SR-TE IPv4 LSP and OSPF SR-TE IPv4 LSP
- IS-IS SR-TE IPv6 LSP
- SR-ISIS IPv4 tunnel stitched to an LDP IPv4 FEC
- BGP IPv4 LSP or BGP IPv6 LSP (with an IPv4 or an IPv4-mapped-IPv6 next-hop) resolved over an SR-ISIS IPv4 tunnel, an SR-OSPF IPv4 tunnel, or an SR-TE IPv4 LSP. This includes support for BGP LSP across AS boundaries and for ECMP next-hops at the transport tunnel level.
- BGP IPv4 LSP (with an IPv6 next-hop) or a BGP IPv6 LSP resolved over an SR-ISIS IPv6 tunnel, an SR-OSPF3 IPv6 tunnel, or an SR-TE IPv6 LSP; including support for BGP LSP across AS boundaries and for ECMP next-hops at the transport tunnel level.
- SR-ISIS or SR-OSPF IPv4 tunnel resolved over IGP IPv4 shortcuts using RSVP-TE LSPs
- SR-ISIS IPv6 tunnel resolved over IGP IPv4 shortcuts using RSVP-TE LSPs
- LDP IPv4 FEC resolved over IGP IPv4 shortcuts using SR-TE LSPs

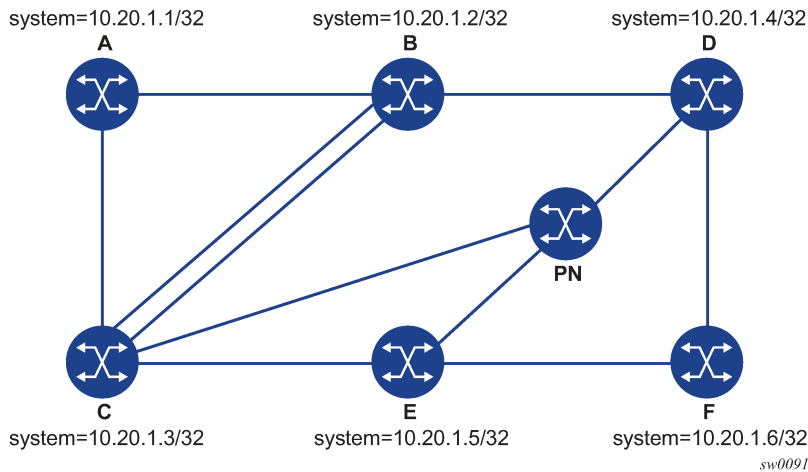
### 3.1.7.2 Operation on SR-ISIS or SR-OSPF tunnels

The following operations apply to **Isp-ping** and **Isp-trace**:

- The sender node builds the Target FEC Stack TLV with a single FEC element corresponding to the node SID of the destination of the SR-ISIS or SR-OSPF tunnel.
- A node SID label that is swapped at an LSR results in the return code of 8, "Label switched at stack-depth <RSC>" as per RFC 8029.
- A node SID label that is popped at an LSR results in a return code of 3, "Replying router is an egress for the FEC at stack-depth <RSC>".
- The **Isp-trace** command is supported with the inclusion of the DSMAP TLV, the DDMAP TLV, or none (when **none** is configured, no Map TLV is sent). The downstream interface information is returned along with the egress label for the node SID tunnel and the protocol that resolved the node SID at the responder node.

[Figure 26: Testing MPLS OAM with SR tunnels](#) shows an example topology for an **Isp-ping** and **Isp-trace** for SR-ISIS node SID tunnel.

Figure 26: Testing MPLS OAM with SR tunnels



Considering this topology, the following is an output example for LSP-PING on DUT-A for target Node SID of DUT-F:

```
*A:Dut-A# oam lsp-ping sr-isis prefix 10.20.1.6/32 igp-instance 0 detail
LSP-PING 10.20.1.6/32: 80 bytes MPLS payload
Seq=1, send from intf int to_B, reply from 10.20.1.6
  udp-data-len=32 ttl=255 rtt=1220324ms rc=3 (EgressRtr)
---- LSP 10.20.1.6/32 PING Statistics ----
1 packets sent, 1 packets received, 0.00% packet loss
round-trip min = 1220324ms, avg = 1220324ms, max = 1220324ms, stddev = 0.000ms
```

The following is an output example for LSP-TRACE on DUT-A for target node SID of DUT-F (DSMAP TLV):

```
*A:Dut-A# oam lsp-trace sr-isis prefix 10.20.1.6/32 igp-instance 0 detail
lsp-trace to 10.20.1.6/32: 0 hops min, 0 hops max, 108 byte packets
1 10.20.1.2 rtt=1220323ms rc=8(DSRtrMatchLabel) rsc=1
  DS 1: ipaddr=10.10.4.4 ifaddr=10.10.4.4 iftype=ipv4Numbered MRU=1496
  label[1]=26406 protocol=6(ISIS)
2 10.20.1.4 rtt=1220323ms rc=8(DSRtrMatchLabel) rsc=1
  DS 1: ipaddr=10.10.9.6 ifaddr=10.10.9.6 iftype=ipv4Numbered MRU=1496
  label[1]=26606 protocol=6(ISIS)
3 10.20.1.6 rtt=1220324ms rc=3(EgressRtr) rsc=1
```

The following is an output example for LSP-TRACE on DUT-A for target node SID of DUT-F (DDMAP TLV):

```
*A:Dut-A# oam lsp-trace sr-isis prefix 10.20.1.6/32 igp-instance 0 downstream-map-tlv ddmapp
detail
lsp-trace to 10.20.1.6/32: 0 hops min, 0 hops max, 108 byte packets
1 10.20.1.2 rtt=1220323ms rc=8(DSRtrMatchLabel) rsc=1
  DS 1: ipaddr=10.10.4.4 ifaddr=10.10.4.4 iftype=ipv4Numbered MRU=1496
  label[1]=26406 protocol=6(ISIS)
2 10.20.1.4 rtt=1220324ms rc=8(DSRtrMatchLabel) rsc=1
  DS 1: ipaddr=10.10.9.6 ifaddr=10.10.9.6 iftype=ipv4Numbered MRU=1496
  label[1]=26606 protocol=6(ISIS)
3 10.20.1.6 rtt=1220324ms rc=3(EgressRtr) rsc=1
```



### 3.1.7.3 Operation on SR-TE LSP

The following operations apply to **lsp-ping** and **lsp-trace**.

- The sender node builds a target FEC Stack TLV that contains FEC elements.
 

For **lsp-ping**, the Target FEC Stack TLV contains a single FEC element that corresponds to the last segment; that is, a node SID or an adjacency SID of the destination of the SR-TE LSP.

For **lsp-trace**, the Target FEC Stack TLV contains a FEC element for each node SID and for each adjacency SID in the path of the SR-TE LSP, including that of the destination of the SR-TE LSP.
- A node SID label popped at an LSR results in a return code of 3 "Replying router is an egress for the FEC at stack-depth <RSC>".
 

An adjacency SID label popped at an LSR results in a return code of 3, "Replying router is an egress for the FEC at stack-depth <RSC>".
- A node SID label that is swapped at an LSR results in the return code of 8, "Label switched at stack-depth <RSC>" per RFC 8029, *Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures*.
 

An adjacency SID label that is swapped at an LSR results in the return code of 8, "Label switched at stack-depth <RSC>" per RFC 8029; for example, in SR OS, "rc=8(DSRtrMatchLabel) rsc=1".
- The **lsp-trace** command is supported with the inclusion of the DSMAP TLV, the DDMAP TLV, or none (when **none** is configured, no Map TLV is sent). The downstream interface information is returned along with the egress label for the node SID tunnel or the adjacency SID tunnel of the current segment as well as the protocol which resolved the tunnel at the responder node.
- When the Target FEC Stack TLV contains more than one FEC element, the responder node that is the termination of one node or adjacency SID segment SID pops its own SID in the first operation. When the sender node receives this reply, it adjusts the Target FEC Stack TLV by stripping the top FEC before sending the probe for the next TTL value. When the responder node receives the next echo request message with the same TTL value from the sender node for the next node SID or adjacency SID segment in the stack, it performs a swap operation to that next segment.
- When the path of the SR-TE LSP is computed by the sender node, the hop-to-label translation tool returns the IGP instance that was used to determine the labels for each hop of the path. When the path of an SR-TE LSP is computed by a PCE, the protocol ID is not returned in the SR-ERO by PCEP. In this case, the sender node performs a lookup in the SR module for the IGP instance that resolved the first segment of the path. In both cases, the determined IGP is used to encode the Protocol ID field of the node SID or adjacency SID in each of the FEC elements of a Target FEC Stack TLV.
- The responder node performs validation of the top FEC in the Target FEC Stack TLV, provided that the depth of the incoming label stack in the packet's header is higher than the depth of the Target FEC Stack TLV.
- TTL values can be changed.

The **ttl** value in **lsp-ping** can be set to a value lower than 255 and the responder node replies if the FEC element in the Target FEC Stack TLV corresponds to a node SID resolved at that node. The responder node, however, fails the validation if the FEC element in the Target FEC Stack TLV is the adjacency of a remote node. The return code in the echo reply message can be one of: "rc=4(NoFECMapping)" or "rc=10(DSRtrUnmatchLabel)".

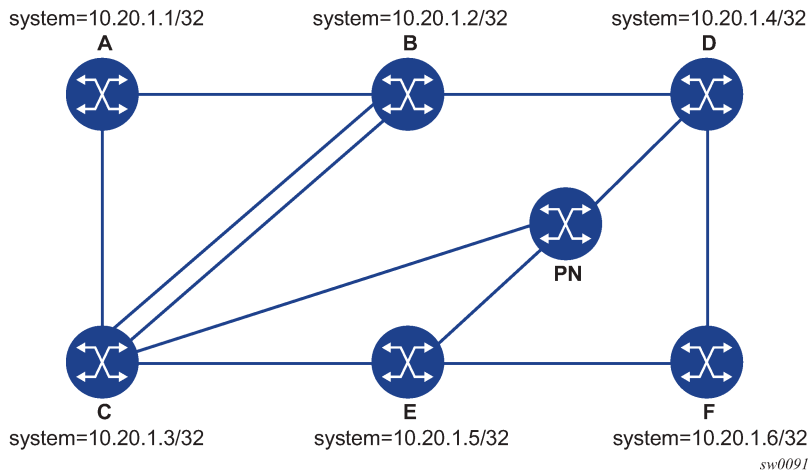
The **min-ttl** and **max-ttl** values in **lsp-trace** can be set to values other than default. The minimum TTL value can, however, properly trace the partial path of an SR-TE LSP only if there is no segment termination before the node that corresponds to the minimum TTL value. Otherwise, it fails validation and returns an error as the responder node would receive a target FEC stack depth that is higher

than the incoming label stack size. The return code in the echo reply message can be one of: "rc=4(NoFECMapping)", "rc=5(DSMappingMismatched)", or "rc=10(DSRtrUnmatchLabel)".

This is true when the **downstream-map-tlv** option is set to any of the **ddmap**, **dsmap**, or **none** values.

The following are example outputs for **lsp-ping** and **lsp-trace** for some SR-TE LSPs. The first one uses a path with strict hops, each corresponding to an adjacency SID, while the second one uses a path with loose hops, each corresponding to a node SID. Assume the topology shown in [Figure 27: Testing MPLS OAM with SR-TE LSP](#).

Figure 27: Testing MPLS OAM with SR-TE LSP



The following is an output example for LSP-PING and LSP-TRACE on DUT-A for strict-hop adjacency SID SR-TE LSP, where:

- source = DUT-A
- destination = DUT-F
- path = A-B, B-C, C-E, E-D, D-F

```
*A:Dut-A# oam lsp-ping sr-te "srteABCDEF" detail
LSP-PING srteABCDEF: 96 bytes MPLS payload
Seq=1, send from intf int_to_B, reply from 10.20.1.6
  udp-data-len=32 ttl=255 rtt=1220325ms rc=3 (EgressRtr)
---- LSP srteABCDEF PING Statistics ----
1 packets sent, 1 packets received, 0.00% packet loss
round-trip min = 1220325ms, avg = 1220325ms, max = 1220325ms, stddev = 0.000ms
*A:Dut-A# oam lsp-trace sr-te "srteABCDEF" downstream-map-tlv ddmap detail
lsp-trace to srteABCDEF: 0 hops min, 0 hops max, 252 byte packets
1 10.20.1.2 rtt=1220323ms rc=3(EgressRtr) rsc=5
1 10.20.1.2 rtt=1220322ms rc=8(DSRtrMatchLabel) rsc=4
  DS 1: ipaddr=10.10.33.3 ifaddr=10.10.33.3 iftype=ipv4Numbered MRU=1520
    label[1]=3 protocol=6(ISIS)
    label[2]=262135 protocol=6(ISIS)
    label[3]=262134 protocol=6(ISIS)
    label[4]=262137 protocol=6(ISIS)
2 10.20.1.3 rtt=1220323ms rc=3(EgressRtr) rsc=4
2 10.20.1.3 rtt=1220323ms rc=8(DSRtrMatchLabel) rsc=3
  DS 1: ipaddr=10.10.5.5 ifaddr=10.10.5.5 iftype=ipv4Numbered MRU=1496
    label[1]=3 protocol=6(ISIS)
    label[2]=262134 protocol=6(ISIS)
    label[3]=262137 protocol=6(ISIS)
3 10.20.1.5 rtt=1220325ms rc=3(EgressRtr) rsc=3
```

```

3 10.20.1.5 rtt=1220325ms rc=8(DSRtrMatchLabel) rsc=2
   DS 1: ipaddr=10.10.11.4 ifaddr=10.10.11.4 iftype=ipv4Numbered MRU=1496
         label[1]=3 protocol=6(ISIS)
         label[2]=262137 protocol=6(ISIS)
4 10.20.1.4 rtt=1220324ms rc=3(EgressRtr) rsc=2
4 10.20.1.4 rtt=1220325ms rc=8(DSRtrMatchLabel) rsc=1
   DS 1: ipaddr=10.10.9.6 ifaddr=10.10.9.6 iftype=ipv4Numbered MRU=1496
         label[1]=3 protocol=6(ISIS)
5 10.20.1.6 rtt=1220325ms rc=3(EgressRtr) rsc=1

```

The following is an output example for LSP-PING and LSP-TRACE on DUT-A for loose-hop Node SID SR-TE LSP, where:

- source = DUT-A
- destination = DUT-F
- path = A, B, C, E

```

*A:Dut-A# oam lsp-ping sr-te "srteABCE_loose" detail
LSP-PING srteABCE_loose: 80 bytes MPLS payload
Seq=1, send from intf int_to_B, reply from 10.20.1.5
   udp-data-len=32 ttl=255 rtt=1220324ms rc=3 (EgressRtr)
---- LSP srteABCE_loose PING Statistics ----
1 packets sent, 1 packets received, 0.00% packet loss
round-trip min = 1220324ms, avg = 1220324ms, max = 1220324ms, stddev = 0.000ms
*A:Dut-A# oam lsp-trace sr-te "srteABCE_loose" downstream-map-tlv ddmmap detail
lsp-trace to srteABCE_loose: 0 hops min, 0 hops max, 140 byte packets
1 10.20.1.2 rtt=1220323ms rc=3(EgressRtr) rsc=3
1 10.20.1.2 rtt=1220322ms rc=8(DSRtrMatchLabel) rsc=2
   DS 1: ipaddr=10.10.3.3 ifaddr=10.10.3.3 iftype=ipv4Numbered MRU=1496
         label[1]=26303 protocol=6(ISIS)
         label[2]=26305 protocol=6(ISIS)
   DS 2: ipaddr=10.10.12.3 ifaddr=10.10.12.3 iftype=ipv4Numbered MRU=1496
         label[1]=26303 protocol=6(ISIS)
         label[2]=26305 protocol=6(ISIS)
   DS 3: ipaddr=10.10.33.3 ifaddr=10.10.33.3 iftype=ipv4Numbered MRU=1496
         label[1]=26303 protocol=6(ISIS)
         label[2]=26305 protocol=6(ISIS)
2 10.20.1.3 rtt=1220323ms rc=3(EgressRtr) rsc=2
2 10.20.1.3 rtt=1220323ms rc=8(DSRtrMatchLabel) rsc=1
   DS 1: ipaddr=10.10.5.5 ifaddr=10.10.5.5 iftype=ipv4Numbered MRU=1496
         label[1]=26505 protocol=6(ISIS)
   DS 2: ipaddr=10.10.11.5 ifaddr=10.10.11.5 iftype=ipv4Numbered MRU=1496
         label[1]=26505 protocol=6(ISIS)
3 10.20.1.5 rtt=1220324ms rc=3(EgressRtr) rsc=1

```

### 3.1.7.4 Operation on an SR-ISIS tunnel stitched to an LDP FEC

The following operations apply to **lsp-ping** and **lsp-trace**:

- The **lsp-ping** tool only works when the responder node is in the same domain (SR or LDP) as the sender node.
- The **lsp-trace** tool works throughout the LDP and SR domains. When used with the DDMAP TLV, **lsp-trace** provides the details of the SR-LDP stitching operation at the boundary node. The boundary node as a responder node replies with the FEC stack change TLV, which contains two operations:
  - a PUSH operation of the SR (LDP) FEC in the LDP-to-SR (SR-to-LDP) direction
  - a POP operation of the LDP (SR) FEC in the LDP-to-SR (SR-to-LDP) direction

- The ICMP tunneling feature is supported for SR-ISIS tunnel stitched to a LDP FEC.

The following is an output example of the **lsp-trace** command with the DDMAP TLV for LDP-to-SR direction (symmetric topology LDP-SR-LDP):

```
*A:Dut-E# oam lsp-trace prefix 10.20.1.2/32 detail downstream-map-tlv ddmmap
lsp-trace to 10.20.1.2/32: 0 hops min, 0 hops max, 108 byte packets
1 10.20.1.3  rtt=3.25ms rc=15(LabelSwitchedWithFecChange) rsc=1
   DS 1: ipaddr=10.10.3.2 ifaddr=10.10.3.2 iftype=ipv4Numbered MRU=1496
       label[1]=26202 protocol=6(ISIS)
       fecchange[1]=POP  fectype=LDP IPv4 prefix=10.20.1.2 remotepeer=0.0.0.0 (Unknown)
       fecchange[2]=PUSH fectype=SR IPv4 Prefix prefix=10.20.1.2 remotepeer=10.10.3.2
2 10.20.1.2  rtt=4.32ms rc=3(EgressRtr) rsc=1
*A:Dut-E#
```

The following is an output example of the **lsp-trace** command with the DDMAP TLV for SR-to-LDP direction (symmetric topology LDP-SR-LDP):

```
*A:Dut-B# oam lsp-trace prefix 10.20.1.5/32 detail downstream-map-tlv ddmmap sr-isis
lsp-trace to 10.20.1.5/32: 0 hops min, 0 hops max, 108 byte packets
1 10.20.1.3  rtt=2.72ms rc=15(LabelSwitchedWithFecChange) rsc=1
   DS 1: ipaddr=10.11.5.5 ifaddr=10.11.5.5 iftype=ipv4Numbered MRU=1496
       label[1]=262143 protocol=3(LDP)
       fecchange[1]=POP  fectype=SR IPv4 Prefix prefix=10.20.1.5 remotepeer=0.0.0.0
(Unknown)
       fecchange[2]=PUSH fectype=LDP IPv4 prefix=10.20.1.5 remotepeer=10.11.5.5
2 10.20.1.5  rtt=4.43ms rc=3(EgressRtr) rsc=1
```

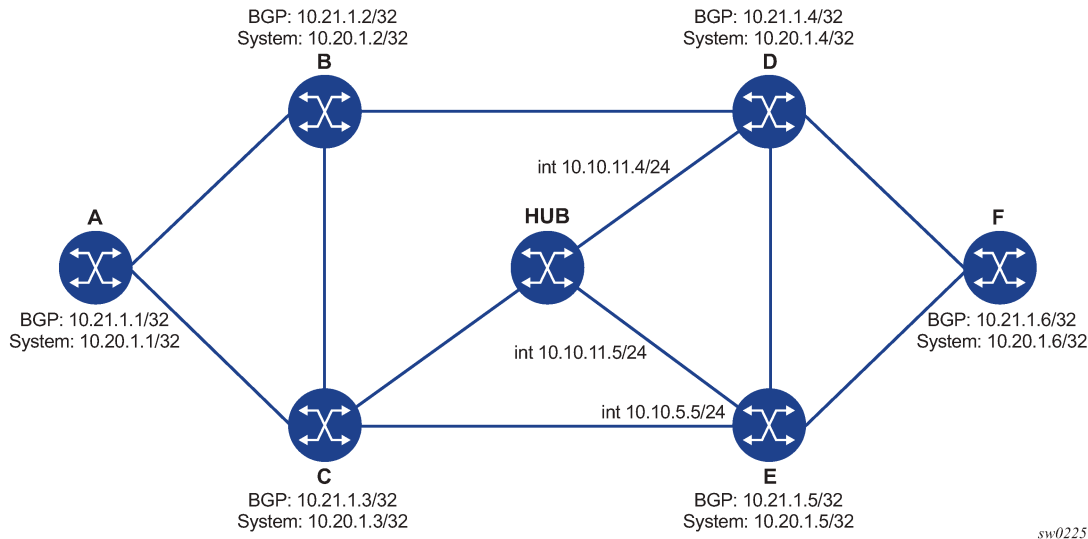
### 3.1.7.5 Operation on a BGP IPv4 LSP resolved over an SR-ISIS IPv4 tunnel, SR-OSPF IPv4 tunnel, or SR-TE IPv4 LSP

SR OS enhances **lsp-ping** and **lsp-trace** of a BGP IPv4 LSP resolved over an SR-ISIS IPv4 tunnel, an SR-OSPF IPv4 tunnel, or an SR-TE IPv4 LSP. The SR OS enhancement reports the full set of ECMP next-hops for the transport tunnel at both ingress PE and at the ABR or ASBR. The list of downstream next-hops is reported in the DSMAP or DDMAP TLV.

When the user initiates an **lsp-trace** of the BGP IPv4 LSP with the **path-destination** option specified, the CPM hash code, at the responder node, selects the outgoing interface to be returned in DSMAP or DDMAP. This decision is based on the modulo operation of the hash value on the label stack or the IP headers (where the DST IP is replaced by the specific 127/8 prefix address in the multipath type 8 field of the DSMAP or DDMAP) of the echo request message and the number of outgoing interfaces in the ECMP set.

[Figure 28: Example topology for BGP over SR-OSPF, SR-TE \(OSPF\), SR-ISIS, and SR-TE \(ISIS\)](#) depicts an example topology used in the subsequent BGP over SR-OSPF, BGP over SR-TE (OSPF), BGP over SR-ISIS, and BGP over SR-TE (ISIS) examples.

Figure 28: Example topology for BGP over SR-OSPF, SR-TE (OSPF), SR-ISIS, and SR-TE (ISIS)



The following are example outputs of the **lsp-trace** command for a hierarchical tunnel consisting of a BGP IPv4 LSP resolved over an SR-ISIS IPv4 tunnel, an SR-OSPF IPv4 tunnel, or an SR-TE IPv4 LSP.

BGP over SR-OSPF example output:

```
*A:Dut-A# oam lsp-trace bgp-label prefix 11.21.1.6/32 detail downstream-map-tlv dmap path-
destination 127.1.1.1
lsp-trace to 11.21.1.6/32: 0 hops min, 0 hops max, 168 byte packets
1 10.20.1.3 rtt=2.31ms rc=8(DSRtrMatchLabel) rsc=2
  DS 1: ipaddr=10.10.5.5 ifaddr=10.10.5.5 iftype=ipv4Numbered MRU=1496
        label[1]=27506 protocol=5(OSPF)
        label[2]=262137 protocol=2(BGP)
  DS 2: ipaddr=10.10.11.4 ifaddr=10.10.11.4 iftype=ipv4Numbered MRU=1496
        label[1]=27406 protocol=5(OSPF)
        label[2]=262137 protocol=2(BGP)
  DS 3: ipaddr=10.10.11.5 ifaddr=10.10.11.5 iftype=ipv4Numbered MRU=1496
        label[1]=27506 protocol=5(OSPF)
        label[2]=262137 protocol=2(BGP)
2 10.20.1.4 rtt=4.91ms rc=8(DSRtrMatchLabel) rsc=2
  DS 1: ipaddr=10.10.9.6 ifaddr=10.10.9.6 iftype=ipv4Numbered MRU=1492
        label[1]=27606 protocol=5(OSPF)
        label[2]=262137 protocol=2(BGP)
3 10.20.1.6 rtt=4.73ms rc=3(EgressRtr) rsc=2
3 10.20.1.6 rtt=5.44ms rc=3(EgressRtr) rsc=1
*A:Dut-A#
```

BGP over SR-TE (OSPF) example output:

```
*A:Dut-A# oam lsp-trace bgp-label prefix 11.21.1.6/32 detail downstream-map-tlv dmap path-
destination 127.1.1.1
lsp-trace to 11.21.1.6/32: 0 hops min, 0 hops max, 236 byte packets
1 10.20.1.2 rtt=2.13ms rc=3(EgressRtr) rsc=4
1 10.20.1.2 rtt=1.79ms rc=8(DSRtrMatchLabel) rsc=3
  DS 1: ipaddr=10.10.4.4 ifaddr=10.10.4.4 iftype=ipv4Numbered MRU=1492
        label[1]=3 protocol=5(OSPF)
        label[2]=262104 protocol=5(OSPF)
        label[3]=262139 protocol=2(BGP)
2 10.20.1.4 rtt=3.24ms rc=3(EgressRtr) rsc=3
```

```

2 10.20.1.4 rtt=4.46ms rc=8(DSRtrMatchLabel) rsc=2
  DS 1: ipaddr=10.10.9.6 ifaddr=10.10.9.6 iftype=ipv4Numbered MRU=1492
        label[1]=3 protocol=5(OSPF)
        label[2]=262139 protocol=2(BGP)
3 10.20.1.6 rtt=6.24ms rc=3(EgressRtr) rsc=2
3 10.20.1.6 rtt=6.18ms rc=3(EgressRtr) rsc=1
*A:Dut-A#

```

#### BGP over SR-ISIS example output:

```

A:Dut-A# oam lsp-trace bgp-label prefix 11.21.1.6/32 detail downstream-map-tlv dmap path-
destination 127.1.1.1
lsp-trace to 11.21.1.6/32: 0 hops min, 0 hops max, 168 byte packets
1 10.20.1.3 rtt=3.33ms rc=8(DSRtrMatchLabel) rsc=2
  DS 1: ipaddr=10.10.5.5 ifaddr=10.10.5.5 iftype=ipv4Numbered MRU=1496
        label[1]=28506 protocol=6(ISIS)
        label[2]=262139 protocol=2(BGP)
  DS 2: ipaddr=10.10.11.4 ifaddr=10.10.11.4 iftype=ipv4Numbered MRU=1496
        label[1]=28406 protocol=6(ISIS)
        label[2]=262139 protocol=2(BGP)
  DS 3: ipaddr=10.10.11.5 ifaddr=10.10.11.5 iftype=ipv4Numbered MRU=1496
        label[1]=28506 protocol=6(ISIS)
        label[2]=262139 protocol=2(BGP)
2 10.20.1.4 rtt=5.12ms rc=8(DSRtrMatchLabel) rsc=2
  DS 1: ipaddr=10.10.9.6 ifaddr=10.10.9.6 iftype=ipv4Numbered MRU=1492
        label[1]=28606 protocol=6(ISIS)
        label[2]=262139 protocol=2(BGP)
3 10.20.1.6 rtt=8.41ms rc=3(EgressRtr) rsc=2
3 10.20.1.6 rtt=6.93ms rc=3(EgressRtr) rsc=1

```

#### BGP over SR-TE (ISIS) example output:

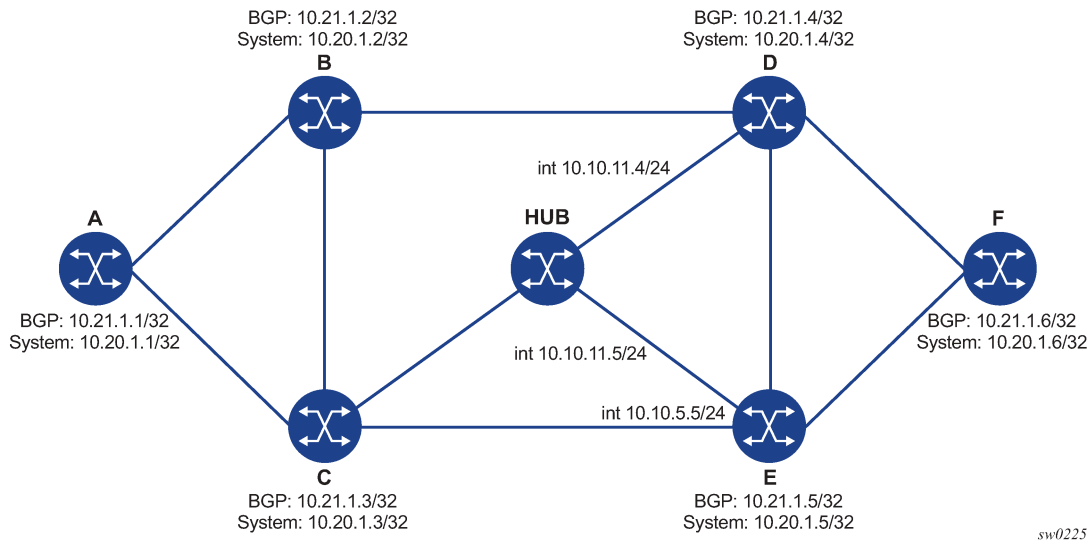
```

*A:Dut-A# oam lsp-trace bgp-label prefix 11.21.1.6/32 detail downstream-map-tlv dmap path-
destination 127.1.1.1
lsp-trace to 11.21.1.6/32: 0 hops min, 0 hops max, 248 byte packets
1 10.20.1.2 rtt=2.60ms rc=3(EgressRtr) rsc=4
1 10.20.1.2 rtt=2.29ms rc=8(DSRtrMatchLabel) rsc=3
  DS 1: ipaddr=10.10.4.4 ifaddr=10.10.4.4 iftype=ipv4Numbered MRU=1492
        label[1]=3 protocol=6(ISIS)
        label[2]=262094 protocol=6(ISIS)
        label[3]=262139 protocol=2(BGP)
2 10.20.1.4 rtt=4.04ms rc=3(EgressRtr) rsc=3
2 10.20.1.4 rtt=4.38ms rc=8(DSRtrMatchLabel) rsc=2
  DS 1: ipaddr=10.10.9.6 ifaddr=10.10.9.6 iftype=ipv4Numbered MRU=1492
        label[1]=3 protocol=6(ISIS)
        label[2]=262139 protocol=2(BGP)
3 10.20.1.6 rtt=6.64ms rc=3(EgressRtr) rsc=2
3 10.20.1.6 rtt=5.94ms rc=3(EgressRtr) rsc=1

```

Assuming the topology in [Figure 29: Example topology for BGP over SR-ISIS in inter-AS option C and BGP over SR-TE \(ISIS\) in inter-AS option C](#) has the addition of an External Border Gateway Protocol (eBGP) peering between nodes B and C, the BGP IPv4 LSP spans the AS boundary and resolves to an SR-ISIS tunnel or an SR-TE LSP within each AS.

Figure 29: Example topology for BGP over SR-ISIS in inter-AS option C and BGP over SR-TE (ISIS) in inter-AS option C



#### BGP over SR-ISIS in inter-AS option C example output:

```
*A:Dut-A# oam lsp-trace bgp-label prefix 11.20.1.6/32 src-ip-address 11.20.1.1 detail
downstream-map-tlv ddmmap path-destination 127.1.1.1
lsp-trace to 11.20.1.6/32: 0 hops min, 0 hops max, 168 byte packets
1 10.20.1.2 rtt=2.69ms rc=3(EgressRtr) rsc=2
1 10.20.1.2 rtt=3.15ms rc=8(DSRtrMatchLabel) rsc=1
  DS 1: ipaddr=10.10.3.3 ifaddr=10.10.3.3 iftype=ipv4Numbered MRU=0
    label[1]=262127 protocol=2(BGP)
2 10.20.1.3 rtt=5.26ms rc=15(LabelSwitchedWithFecChange) rsc=1
  DS 1: ipaddr=10.10.5.5 ifaddr=10.10.5.5 iftype=ipv4Numbered MRU=1496
    label[1]=26506 protocol=6(ISIS)
    label[2]=262139 protocol=2(BGP)
    fecchange[1]=PUSH fectype=SR Ipv4 Prefix prefix=10.20.1.6 remotepeer=10.10.5.5
3 10.20.1.5 rtt=7.08ms rc=8(DSRtrMatchLabel) rsc=2
  DS 1: ipaddr=10.10.10.6 ifaddr=10.10.10.6 iftype=ipv4Numbered MRU=1496
    label[1]=26606 protocol=6(ISIS)
    label[2]=262139 protocol=2(BGP)
4 10.20.1.6 rtt=9.41ms rc=3(EgressRtr) rsc=2
4 10.20.1.6 rtt=9.53ms rc=3(EgressRtr) rsc=1
```

#### BGP over SR-TE (ISIS) in inter-AS option C example output:

```
*A:Dut-A# oam lsp-trace bgp-label prefix 11.20.1.6/32 src-ip-address 11.20.1.1 detail
downstream-map-tlv ddmmap path-destination 127.1.1.1
lsp-trace to 11.20.1.6/32: 0 hops min, 0 hops max, 168 byte packets
1 10.20.1.2 rtt=2.77ms rc=3(EgressRtr) rsc=2
1 10.20.1.2 rtt=2.92ms rc=8(DSRtrMatchLabel) rsc=1
  DS 1: ipaddr=10.10.3.3 ifaddr=10.10.3.3 iftype=ipv4Numbered MRU=0
    label[1]=262127 protocol=2(BGP)
2 10.20.1.3 rtt=4.82ms rc=15(LabelSwitchedWithFecChange) rsc=1
  DS 1: ipaddr=10.10.5.5 ifaddr=10.10.5.5 iftype=ipv4Numbered MRU=1496
    label[1]=26505 protocol=6(ISIS)
    label[2]=26506 protocol=6(ISIS)
    label[3]=262139 protocol=2(BGP)
    fecchange[1]=PUSH fectype=SR Ipv4 Prefix prefix=10.20.1.6 remotepeer=
0.0.0.0 (Unknown)
```

```

fecchange[2]=PUSH fectype=SR Ipv4 Prefix prefix=10.20.1.5          remotepeer=
10.10.5.5
3 10.20.1.5  rtt=7.10ms rc=3(EgressRtr) rsc=3
3 10.20.1.5  rtt=7.45ms rc=8(DSRtrMatchLabel) rsc=2
   DS 1: ipaddr=10.10.10.6 ifaddr=10.10.10.6 iftype=ipv4Numbered MRU=1496
       label[1]=26606 protocol=6(ISIS)
       label[2]=262139 protocol=2(BGP)
4 10.20.1.6  rtt=9.23ms c=3(EgressRtr) rsc=2
4 10.20.1.6  rtt=9.46ms rc=3(EgressRtr) rsc=1
*A:Dut-A

```

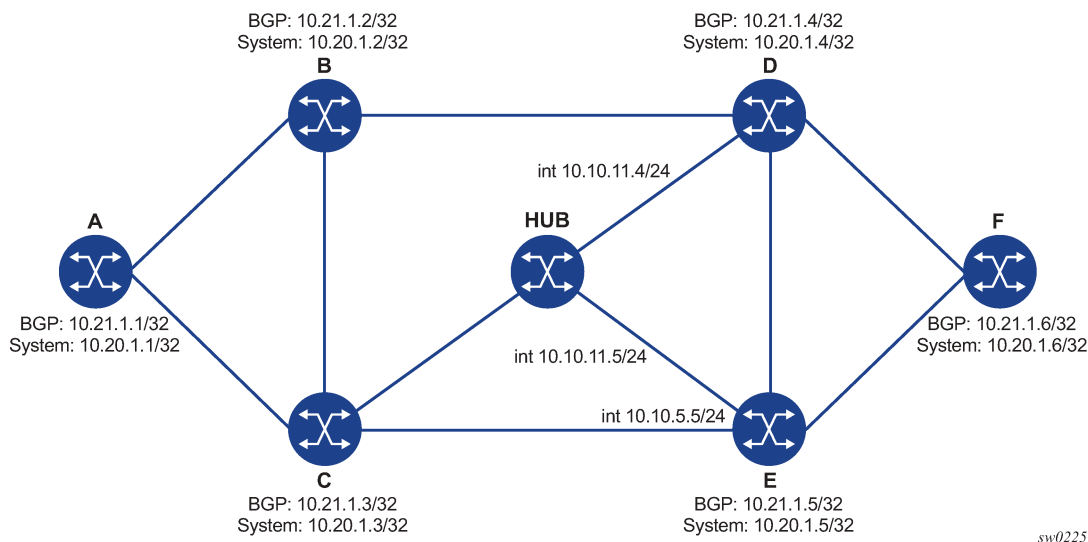
### 3.1.7.6 Operation on an SR-ISIS IPv4 tunnel, IPv6 tunnel, or SR-OSPF IPv4 tunnel resolved over IGP IPv4 shortcuts using RSVP-TE LSPs

When IGP shortcut is enabled in an IS-IS or an OSPF instance and the family SRv4 or SRv6 is set to resolve over RSVP-TE LSPs, a hierarchical tunnel is created whereby an SR-ISIS IPv4 tunnel, an SR-ISIS IPv6 tunnel, or an SR-OSPF tunnel resolves over the IGP IPv4 shortcuts using RSVP-TE LSPs.

The following example outputs are of the **lsp-trace** command for a hierarchical tunnel consisting of an SR-ISIS IPv4 tunnel and an SR-OSPF IPv4 tunnel, resolving over an IGP IPv4 shortcut using a RSVP-TE LSP.

The topology, as shown in [Figure 30: Example topology for SR-ISIS over RSVP-TE and SR-OSPF over RSVP-TE](#), is used for the following SR-ISIS over RSVP-TE and SR-OSPF over RSVP-TE example outputs.

Figure 30: Example topology for SR-ISIS over RSVP-TE and SR-OSPF over RSVP-TE



SR-ISIS over RSVP-TE example output:

```

*A:Dut-F# oam lsp-trace sr-isis prefix 10.20.1.1/32 detail path-destination 127.1.1.1 igp-
instance 1
lsp-trace to 10.20.1.1/32: 0 hops min, 0 hops max, 180 byte packets
1 10.20.1.4  rtt=5.05ms rc=8(DSRtrMatchLabel) rsc=2
   DS 1: ipaddr=10.10.4.2 ifaddr=10.10.4.2 iftype=ipv4Numbered MRU=1500
       label[1]=262121 protocol=4(RSVP-TE)
       label[2]=28101 protocol=6(ISIS)
2 10.20.1.2  rtt=5.56ms rc=8(DSRtrMatchLabel) rsc=2

```



```

DS 1: ipaddr=10.10.1.1 ifaddr=10.10.1.1 iftype=ipv4Numbered MRU=1500
      label[1]=262124 protocol=4(RSVP-TE)
      label[2]=28101 protocol=6(ISIS)
3 10.20.1.1 rtt=7.30ms rc=3(EgressRtr) rsc=2
3 10.20.1.1 rtt=5.40ms rc=3(EgressRtr) rsc=1
*A:Dut-F#
    
```

SR-OSPF over RSVP-TE example output:

```

*A:Dut-F# oam lsp-trace sr-ospf prefix 10.20.1.1/32 detail path-destination 127.1.1.1 igp-
instance 2
lsp-trace to 10.20.1.1/32: 0 hops min, 0 hops max, 180 byte packets
1 10.20.1.4 rtt=3.24ms rc=8(DSRtrMatchLabel) rsc=2
  DS 1: ipaddr=10.10.4.2 ifaddr=10.10.4.2 iftype=ipv4Numbered MRU=1500
      label[1]=262125 protocol=4(RSVP-TE)
      label[2]=27101 protocol=5(OSPF)
2 10.20.1.2 rtt=5.77ms rc=8(DSRtrMatchLabel) rsc=2
  DS 1: ipaddr=10.10.1.1 ifaddr=10.10.1.1 iftype=ipv4Numbered MRU=1500
      label[1]=262124 protocol=4(RSVP-TE)
      label[2]=27101 protocol=5(OSPF)
3 10.20.1.1 rtt=7.19ms rc=3(EgressRtr) rsc=2
3 10.20.1.1 rtt=8.41ms rc=3(EgressRtr) rsc=1
    
```

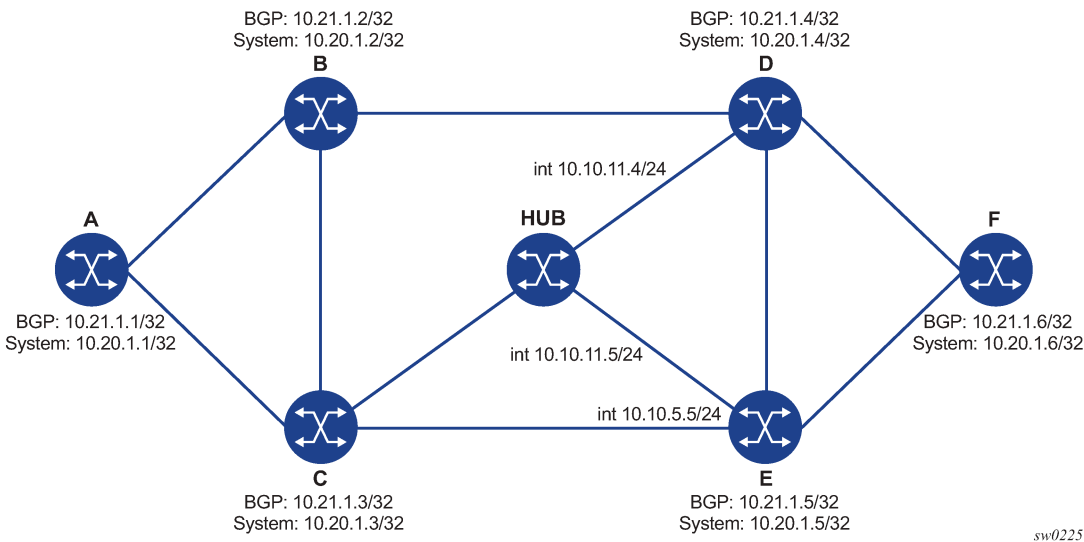
### 3.1.7.7 Operation on an LDP IPv4 FEC resolved over IGP IPv4 shortcuts using SR-TE LSPs

When IGP shortcut is enabled in an IS-IS or an OSPF instance and the family IPv4 is set to resolve over SR-TE LSPs, a hierarchical tunnel is created whereby an LDP IPv4 FEC resolves over the IGP IPv4 shortcuts using SR-TE LSPs.

The following are example outputs of the **lsp-trace** command for a hierarchical tunnel consisting of a LDP IPv4 FEC resolving over a IGP IPv4 shortcut using a SR-TE LSP.

The topology, as shown in [Figure 31: Example topology for LDP over SR-TE \(ISIS\) and LDP over SR-TE \(OSPF\)](#), is used for the following LDP over SR-TE (ISIS) and LDP over SR-TE (OSPF) example outputs.

Figure 31: Example topology for LDP over SR-TE (ISIS) and LDP over SR-TE (OSPF)



## LDP over SR-TE (ISIS) example output:

```

*A:Dut-F# oam lsp-trace prefix 10.20.1.1/32 detail path-destination 127.1.1.1
lsp-trace to 10.20.1.1/32: 0 hops min, 0 hops max, 184 byte packets
1 10.20.1.4 rtt=2.33ms rc=8(DSRtrMatchLabel) rsc=3
   DS 1: ipaddr=10.10.4.2 ifaddr=10.10.4.2 iftype=ipv4Numbered MRU=1492
       label[1]=28202 protocol=6(ISIS)
       label[2]=28201 protocol=6(ISIS)
       label[3]=262138 protocol=3(LDP)
2 10.20.1.2 rtt=6.39m rc=3(EgressRtr) rsc=3
2 10.20.1.2 rtt=7.29ms rc=8(DSRtrMatchLabel) rsc=2
   DS 1: ipaddr=10.10.1.1 ifaddr=10.10.1.1 iftype=ipv4Numbered MRU=1492
       label[1]=28101 protocol=6(ISIS)
       label[2]=262138 protocol=3(LDP)
3 10.20.1.1 rtt=8.34m rc=3(EgressRtr) rsc=2
3 10.20.1.1 rtt=9.37ms rc=3(EgressRtr) rsc=1

*A:Dut-F# oam lsp-ping prefix 10.20.1.1/32 detail
LSP-PING 10.20.1.1/32: 80 bytes MPLS payload
Seq=1, send from intf int_to_D, reply from 10.20.1.1
  udp-data-len=32 ttl=255 rtt=8.21ms rc=3 (EgressRtr)
---- LSP 10.20.1.1/32 PING Statistics ----
1 packets sent, 1 packets received, 0.00% packet loss
round-trip mi = 8.21ms, avg = 8.21ms, max = 8.21ms, stddev = 0.000ms
=====
LDP Bindings (IPv4 LSR ID 10.20.1.6)
(IPv6 LSR ID fc00::a14:106)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch, BA - ASBR Backup FEC
=====
LDP IPv4 Prefix Bindings
=====
Prefix                               IngLbl                               EgrLbl
Peer                                  EgrIntf/LspId
EgrNextHop
-----
10.20.1.1/32                          --                                   262138
10.20.1.1:0                            LspId 655467
10.20.1.1
10.20.1.1/32                          262070U                             262040
10.20.1.3:0                            --
--
10.20.1.1/32                          262070U                             --
10.20.1.4:0                            --
--
10.20.1.1/32                          262070U                             262091
10.20.1.5:0                            --
--
10.20.1.1/32                          --                                   262138
fc00::a14:101[0]                       --
--
10.20.1.1/32                          262070U                             262040
fc00::a14:103[0]                       --
--

```

```

10.20.1.1/32                262070U                262091
fc00::a14:105[0]          --
--

```

```

-----
No. of IPv4 Prefix Bindings: 7
=====

```

#### LDP over SR-TE (OSPF) example output:

```

*A:Dut-F# oam lsp-trace prefix 10.20.1.1/32 detail path-destination 127.1.1.1
lsp-trace to 10.20.1.1/32: 0 hops min, 0 hops max, 184 byte packets
1 10.20.1.4 rtt=2.73ms rc=8(DSRtrMatchLabel) rsc=3
   DS 1: ipaddr=10.10.4.2 ifaddr=10.10.4.2 iftype=ipv4Numbered MRU=1492
        label[1]=27202 protocol=5(OSPF)
        label[2]=27201 protocol=5(OSPF)
        label[3]=262143 protocol=3(LDP)
2 10.20.1.2 rtt=6.77ms rc=3(EgressRtr) rsc=3
2 10.20.1.2 rtt=6.75ms rc=8(DSRtrMatchLabel) rsc=2
   DS 1: ipaddr=10.10.1.1 ifaddr=10.10.1.1 iftype=ipv4Numbered MRU=1492
        label[1]=27101 protocol=5(OSPF)
        label[2]=262143 protocol=3(LDP)
3 10.20.1.1 rtt=7.10ms rc=3(EgressRtr) rsc=2
3 10.20.1.1 rtt=7.53ms rc=3(EgressRtr) rsc=1

*A:Dut-F# oam lsp-ping prefix 10.20.1.1/32 detail
LSP-PING 10.20.1.1/32: 80 bytes MPLS payload
Seq=1, send from intf int_to_D, reply from 10.20.1.1
   udp-data-len=32 ttl=255 rtt=8.09ms rc=3 (EgressRtr)

---- LSP 10.20.1.1/32 PING Statistics ----
1 packets sent, 1 packets received, 0.00% packet loss
round-trip min = 8.09ms, avg = 8.09ms, max = 8.09ms, stddev = 0.000ms

=====
LDP Bindings (IPv4 LSR ID 10.20.1.6)
              (IPv6 LSR ID fc00::a14:106)
=====
Label Status:
  U - Label In Use, N - Label Not In Use, W - Label Withdrawn
  WP - Label Withdraw Pending, BU - Alternate For Fast Re-Route
  e - Label ELC
FEC Flags:
  LF - Lower FEC, UF - Upper FEC, M - Community Mismatch, BA - ASBR Backup FEC
=====
LDP IPv4 Prefix Bindings
=====
Prefix                IngLbl                EgrLbl
Peer                  EgrIntf/LspId
EgrNextHop
-----
10.20.1.1/32          --                    262143
10.20.1.1:0          LspId 655467
10.20.1.1
10.20.1.1/32          262089U               262135
10.20.1.3:0          --
--

```

```

10.20.1.1/32                262089U                --
10.20.1.4:0                --
--
10.20.1.1/32                262089U                262129
10.20.1.5:0                --
--
10.20.1.1/32                --                        262143
fc00::a14:101[0]          --
--
10.20.1.1/32                262089U                262135
fc00::a14:103[0]          --
--
10.20.1.1/32                262089U                262129
fc00::a14:105[0]          --
--
-----
No. of IPv4 Prefix Bindings: 7
=====

```

### 3.1.8 MPLS OAM support in IPv4 or IPv6 SR policies

This feature extends the support of `lsp-ping`, `lsp-trace`, and ICMP tunneling probes to IPv4 and IPv6 SR policies.

This feature describes the CLI options for `lsp-ping` and `lsp-trace` commands under the OAM and SAA contexts for the following type of Segment Routing tunnel: **sr-policy**.

- **oam lsp-ping sr-policy** {color integer <0..4294967295> endpoint ip-address<ipv4/ipv6>} [segment-list id<1..32>] [src-ip-address ip-address] [fc fc-name [profile {in|out}]] [size octets] [ttl label-ttl] [send-count send-count] [timeout timeout] [interval interval] [path-destination ip-address [interface if-name | next-hop ip-address]] [detail]
- **oam lsp-trace sr-policy** {color integer <0..4294967295> endpoint ip-address<ipv4/ipv6>} [segment-list id<1..32>] [src-ip-address ip-address] [fc fc-name [profile {in|out}]] [max-fail no-response-count] [probe-count probes-per-hop] [size octets] [min-ttl min-label-ttl] [max-ttl max-label-ttl] [timeout timeout] [interval interval] [path-destination ip-address [interface if-name | next-hop ip-address]] [downstream-map-tlv {dsmap | ddmmap | none}] [detail]

The CLI does not require entry of the SR policy **head-end** parameter that corresponds to the IPv4 address of the router where the static SR policy is configured or where the BGP NRLRI of the SR policy is sent to by a controller or another BGP speaker. SR OS expects its IPv4 system address in the **head-end** parameter of both the IPv4 and IPv6 SR policy NLRIs, otherwise SR OS does not import the NRLI.

The source IPv4 or IPv6 address can be specified to encode in the Echo Request message of the LSP ping or LSP trace packet.

The **endpoint** command specifies the endpoint of the policy and which can consist of an IPv4 address, and therefore matching to a SR policy in the IPv4 tunnel-table, or an IPv6 address and therefore matching to a SR policy in the IPv6 tunnel-table.

The **color** command must correspond to the SR policy color attribute that is configured locally in the case of a static policy instance or signaled in the NLRI of the BGP signaled SR policy instance.

The **endpoint** and **color** commands test the active path (or instance) of the identified SR policy only.

The **lsp-ping** and **lsp-trace** commands can test one segment list at a time by specifying one segment list of the active instance of the policy or active candidate path. In this case, the **segment-list id** command is configured or segment list 1 is tested by default. The segment-list ID corresponds to the same index that was used to save the SR policy instance in the SR policy database. In the case of a static SR policy, the segment-list ID matches the segment list index entered in the configuration. In both the static and the BGP SR policies, the segment-list ID matches the index displayed for the segment list in the output of the **show** command of the policies.

The exercised segment list corresponds to a single SR-TE path with its own NHLFE or super NHLFE in the data path.

The ICMP tunneling feature support with SR policy is described in [ICMP-tunneling operation](#) and does not require additional CLI commands.

### 3.1.8.1 LSP ping and LSP trace operation

The following operations are supported with both **lsp-ping** and **lsp-trace**.

- The **lsp-ping** and **lsp-trace** features model the tested segment list as a NIL FEC target FEC stack.
- Both an IPv4 SR policy (endpoint is an IPv4 address) and IPv6 SR policy (endpoint is an IPv6 address) can potentially contain a mix of IPv4 and IPv6 (node, adjacency, or adjacency set) SIDs in the same segment list or across segment lists of the same policy. While this is not a typical use of the SR policy, it is nonetheless allowed in the IETF standard and supported in SR OS. As a result, the downstream interface and node address information returned in the DSMAP or DDMAP TLV can have a different IP family across the path of the SR policy.

Also, the IPv4 or IPv6 endpoint address can be null (0.0.0.0 or 0::0). This has no impact on the OAM capability.

- Unlike a SR-TE LSP path, the type of each segment (node, adjacency, or adjacency set) in the SID list may not be known to the sender node, except for the top SID that is validated by the SR policy database and which uses this segment type to resolve the outgoing interface or interfaces and outgoing label or labels to forward the packet out.
- The NIL FEC type is used to represent each SID in the segment list, including the top SID. The NIL FEC is defined RFC 8029 and has three main applications:

- Allow the sender node to insert a FEC stack sub-TLV into the target FEC TLV when the FEC type is not known to the sender node (case of SIDs of the SR policy except the top SID) or if there is no explicit FEC associated with the label (case of a label of a static LSP or a MPLS forwarding policy). This is the application applicable to the SR policy.

Although the sender node knows the FEC type for the top SID in the segment list of a SR policy, the NIL FEC is used for consistency. However, the sender node does all the processing required to look up the top SID as per the procedures of any other explicit FEC type.

- Allow the sender node to insert a FEC stack sub-TLV into the target FEC stack sub-TLV if a special purpose label (for example, Router Alert) is inserted in the packet's label stack to maintain the correct 1-to-1 mapping of the packet's stacked labels to the hierarchy of FEC elements in the target FEC stack TLV processing at the responder node.

SR OS does not support this application in a sender node role but can process the NIL FEC if received by a third-party implementation.

- Allow the responder node to hide from the sender node a FEC element that it is pushing or stitching to by adding a NIL FEC TLV with a PUSH or a POP and PUSH (equivalent to a SWAP) operation into the FEC stack change sub-TLV.

SR OS does not support this application in a sender node role but can process the NIL FEC if received by a third-party implementation.

- In the case of **lsp-ping**, the sender node builds a target FEC Stack TLV which contains a single NIL FEC element corresponding to the last segment of the tested segment list of the SR policy.
- In the case of **lsp-trace**, the sender node builds a target FEC Stack TLV which contains a NIL FEC element for each SID in the segment list.
- To support the processing of the NIL FEC in the context of the SR policy and the applications in RFC 8029, SR OS in a receiver node role performs the following operations:
  1. Look-up the label of the Nil FEC in the SR database to match against the SID of a resolved node, a resolved adjacency, a resolved adjacency SET or a binding SID.
  2. If a match exists, continue processing of the NIL FEC.
  3. Otherwise, look up the label of the NIL FEC in the Label Manager.
  4. If a match exists, then process the FEC as per the POP or SWAP operation provided by the lookup and following the NIL FEC procedures in RFC 8029.
  5. Otherwise, fail the validation and send a return code of 3 <Replying router has no mapping for the FEC at stack-depth <RSC>> in the MPLS echo reply message. The sender node fails the probe at this point.

- A SID label associated with a NIL FEC and which is popped at an LSR, acting in a receiver node role, is first looked up. If the label is valid, then the processing results in a return code of 3 <Replying router is an egress for the FEC at stack-depth <RSC>>.

A label is valid if the LSR validates it in its Segment Routing (SR) database. Because the LSR does not know the actual FEC type and FEC value, it successfully validates it if the SR database indicates a programmed POP operation with that label for a node SID exists.

- A SID label associated with a NIL FEC and which is swapped at an LSR, acting in a receiver node role, is first looked up. If the label is valid, then the processing results in the return code of 8 Label switched at stack-depth <RSC> as per RFC 8029.

A label is valid if the LSR validates it in its Segment Routing (SR) database. Because the LSR does not know the actual FEC type and FEC value, it successfully validates it if the SR database indicates a programmed SWAP operation with that label for either a node SID, an adjacency SID, an adjacency SET SID, or a binding SID exists.

The swap operation corresponds to swapping the incoming label to an implicit-null label toward the downstream router in the case of an adjacency and toward a set of downstream routers in the case of an adjacency set.

The swap operation corresponds to swapping the incoming label to one or more labels toward a set of downstream routers in the case of a node SID and a binding SID.

- The **lsp-trace** command is supported with the inclusion of the DSMAP TLV, the DDMAP TLV or none of them by the sender node in the ECHO request message. The responder node returns in the DSMAP or DDMAP TLV the downstream interface information along with the egress label and protocol ID that corresponds to the looked up node SID, adjacency SID, adjacency SET SID, or binding SID.
- When the Target FEC Stack TLV contains more than one NIL FEC element, the responder node that is the termination of a FEC element indicates the FEC POP operation implicitly by replying with a return

code of 3 <Replying router is an egress for the FEC at stack-depth <RSC>>. When the sender node gets this reply, the sender node adjusts the Target FEC Stack TLV by stripping the top FEC before sending the next probe for the same TTL value. When the responder node receives the next echo request message with the same TTL value from the sender node, the responder node processes the next FEC element in the stack.

- The responder node performs validation of the top FEC in the target FEC stack TLV provided that the depth of the incoming label stack in the packet's header is strictly higher than the depth of the target FEC stack TLV.
- The **ttl** value in **lsp-ping** context can be set to a value lower than 255 and the responder node replies if the NIL FEC element in the Target FEC Stack TLV corresponds to a node SID resolved at that node. The responder node, however, fails the validation if the NIL FEC element in Target FEC Stack TLV corresponds to adjacency of a remote node. The return code in the echo reply message can be one of: rc=4(NoFECMapping), and rc=10(DSRtrUnmatchLabel).
- The **min-ttl** and **max-ttl** commands in **lsp-trace** context can be set to values other than default. The **min-ttl** can, however, properly trace the partial path of a SR policy only if there is not segment termination before the node that corresponds to the **min-ttl** value. Otherwise, the validation fails and returns an error as the responder node receives a Target FEC Stack depth that is higher than incoming label stack size. The return code in the echo reply message can be one of: rc=4(NoFECMapping), rc=5(DSMappingMismatched), and rc=10(DSRtrUnmatchLabel).

This is true when the **downstream-map-tlv** option is set to any of **ddmap**, **dsmap**, or **none** values.

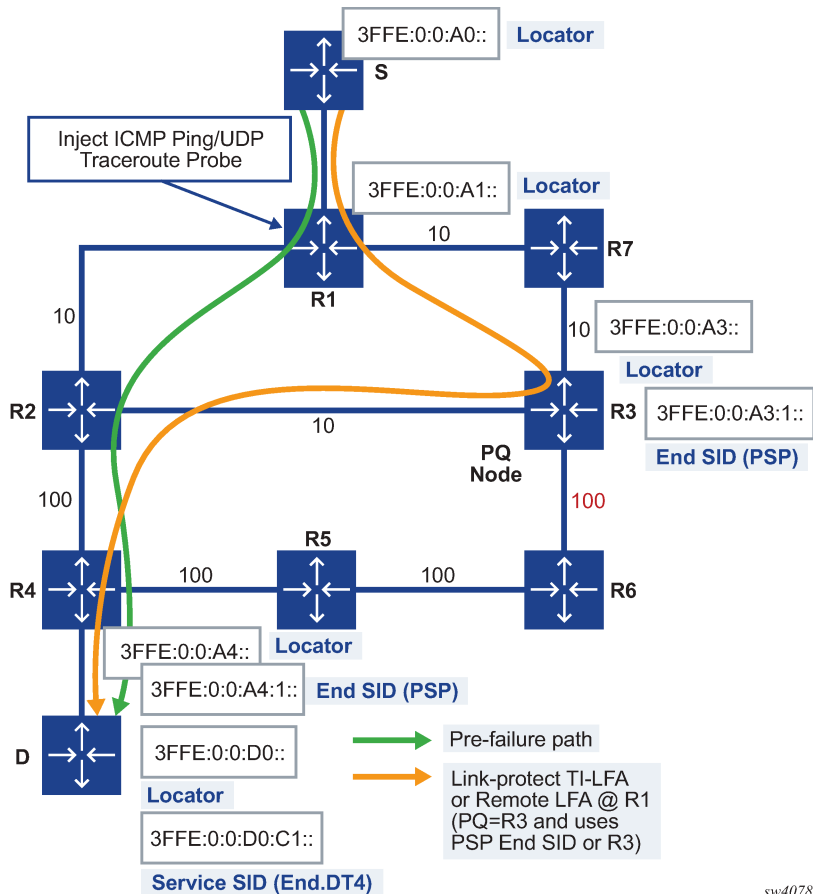
### 3.1.8.2 ICMP-tunneling operation

The ICMP tunneling feature operates in the same way as in a SR-TE LSP. When the label TTL of a traceroute packet of a core IPv4 or IPv6 route or a vpn-ipv4 or vpn-ipv6 route expires at an LSR, the latter generates an ICMP reply packet of type=11- (time exceeded) and injects it in the forward direction of the SR policy. When the packet is received by the egress LER or a BGP border router, SR OS performs a regular user packet route lookup in the data path in the GRT context or in a VPRN context and forwards the packet to the destination. The destination of the packet is the sender of the original packet which TTL expired at the LSR.

### 3.1.9 Segment Routing IPv6 (SRv6) OAM

The network setup illustrated in [Figure 32: SRv6 OAM network setup](#) shows an example configuration of segment routing using SRv6 that is discussed in this section. See the *7750 SR and 7950 XRS Segment Routing and PCE User Guide* for further information about the SRv6 feature.

Figure 32: SRv6 OAM network setup



sw4078

As shown in [Figure 32: SRv6 OAM network setup](#), the network administrator originates a ping or a traceroute probe on node R1 to test the path of an SRv6 locator of node D, an SRv6 segment identifier (SID) owned by node D, or an IP prefix resolved to an SRv6 tunnel toward node D. R1 is referred to as the sender node. Node D is referred to as the target node because it owns the target locator or SID that is being tested. A target node can be any router in the SRv6 network domain which either owns the target locator or SID, or a router in which the OAM probe was extracted because a local route matches or because of the value of the hop-limit field setting in the packet.

The primary path to D is through R2 and R4. The link-protect TI-LFA backup path is through R3 as a PQ node and then R2 and R4.

The classic **ping** and **traceroute** OAM CLI commands are used to test an IPv4 or IPv6 prefix in a virtual routing and forwarding (VRF) table or in the base router table when resolved to an SRv6 tunnel, for example:

```
ping address [detail] [source ip-address]
```

```
traceroute address [detail] source ip-address] decode original-datagram
```

The same CLI commands are used to test the address of an SRv6 locator or a SID. In this case, the user enters the IPv6 address of the target locator prefix or the target SID.

The source address encoded in the outer IPv6 header of the ping or traceroute packet is derived from the following steps, in ascending order:



1. the user-entered source IPv6 address in the **ping** or **traceroute** command, which is checked for validity against a local interface address or a local locator prefix or SID
2. the globally-configured IPv6 source address from the **source-address application6 ping** or **source-address application6 traceroute** commands
3. the preferred primary IPv6 address of the system interface
4. the IPv6 address of outgoing interface

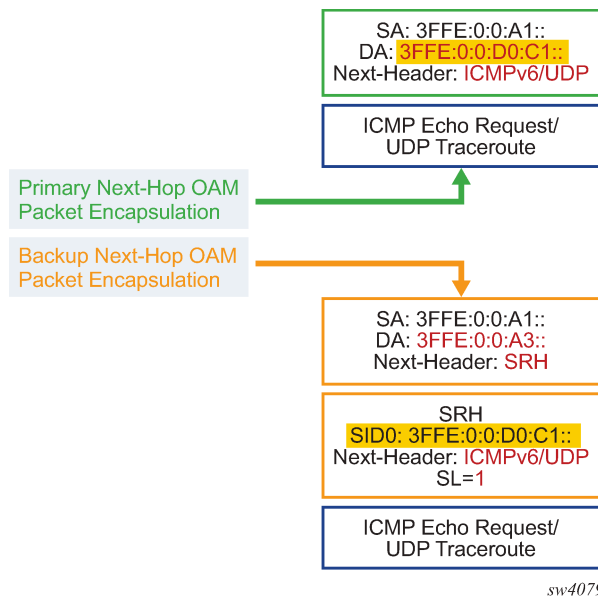
### 3.1.9.1 Ping or traceroute of SRv6 remote locator or remote SID (End, End.X, End.DT4, End.DT6, End.DT46, End.DX2, End.DT2M and End.DT2U )

The features in this section comply to *draft-ietf-6man-spring-srv6-oam, Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)*.

#### 3.1.9.1.1 Ingress PE router (sender node) behavior

The packet is encoded with a destination address set to the remote locator prefix or the specific remote SID, and the next-header field is ICMPv6 (for ping's Echo request message) or UDP (for traceroute). The packet is encapsulated as shown in [Figure 33: Packet encapsulation for ping or traceroute of a remote locator/SID](#). When the Topology-Independent Loop-Free Alternate (TI-LFA) or Remote LFA repair tunnel is activated, the LFA segment routing header (LFA SRH) is also pushed on the encapsulation of the SRv6 tunnel to the node D.

Figure 33: Packet encapsulation for ping or traceroute of a remote locator/SID



The outer IPv6 header hop-limit field is set according to the operation of the probe. For ping, the hop limit uses the default 254 or a user-entered value.

For traceroute, the hop limit is incrementally increased using one of the following:

- from 1 until the packet reaches the egress PE

- from the configured minimum value to the maximum value or until the packet reaches the egress PE

The ingress PE looks up the prefix of the locator or SID in the routing table and if a route exists, it forwards the packet to the next hop. The ingress PE does not check if the target SID or locator has been received in ISIS or BGP.

### 3.1.9.1.2 Transit P router behavior

Ping and traceroute operate similarly to any data or OAM IPv6 packet when expiring (the value in the **hop-limit** field is equal to or less than 1) at a transit SRv6 node, whether this node is a SID termination or not.

The data path at the ingress network interface where the packet is received extracts the packet to the CPM.

The CPM originates a TTL expiry ICMP reply message Type: "Time Exceeded", Code: "Time to Live exceeded in Transit".

The CPM sends the reply to the SRv6 router whose address is encoded in the SA field of the outer IPv6 header in the received packet. The source address is set to the system IPv6 address, if configured, or the address of the interface used to forward the packet to the next hop.

When the transport protocol of the traceroute packet is UDP or TCP (as indicated in the **traceroute** command), the intermediate node copies a portion of the original datagram, up to 1232 bytes, into the reply message payload. This may include the outer IPv6 header and SRH headers. To copy the maximum information from the original datagram, the node must include the **configure test-oam icmp ipv6 maximum-original-datagram** option.

When the **hop-limit** field value is higher than 1, the packet is processed in the data path similar to the process of any SRv6 user data packet in the transit router. See the *7750 SR and 7950 XRS Segment Routing and PCE User Guide* for more information.

### 3.1.9.1.3 Egress PE router (target node) behavior

The data path in the ingress network port in the destination router that owns the target SID extracts the packet to CPM.

A traceroute packet is extracted based on the hop-limit field value of 1 before the route lookup.

A ping packet is extracted after the route lookup matches a FIB entry of a local locator, End, or End.X SID.

The CPM checks that the target locator or SID address matches a local entry. This means that the locator or SID has either been configured manually by the user or it has been auto-allocated by the locator module for use by IS-IS or BGP.

A match on the locator requires an exact match on the locator field and that both the function and argument fields be zero.

A match on a SID requires both the locator and function fields to match. The argument field is not checked.

When a match on a local locator or SID exists, the CPM replies with the following:

- **in the case of ICMPv6 ping**

CPM replies with an ICMPv6 echo reply message.

The source address of the packet is set to the address in the DA field of the packet of the received echo request message.

- **in the case of UDP traceroute**

CPM replies with an ICMPv6 message (Type: "Destination unreachable", Code: "Port Unreachable").

The source address is set to the system IPv6 address if configured or the address of the interface used to forward the packet to the next hop.

The target node copies a portion of the original datagram, up to 1232 bytes, of the received packet into the reply message payload. This may include the outer IPv6 header and any SRH headers. To copy the maximum information from the original datagram, the node must configure the **configure test-oam icmp ipv6 maximum-original-datagram** option.

When the traceroute is carried over a TCP transport and the destination is not an interface address, there is no indication whether the TCP port is open or closed.

Figure 34: End-to-end packet encapsulation for ping or traceroute of a remote locator or SID (1) and Figure 35: End-to-end packet encapsulation for ping or traceroute of a remote locator or SID (2) show the packet encapsulation from ingress PE to egress PE for both the primary path and the backup path. For the backup path, both the PSP and USP types of the LFA SRH are shown.

Figure 34: End-to-end packet encapsulation for ping or traceroute of a remote locator or SID (1)

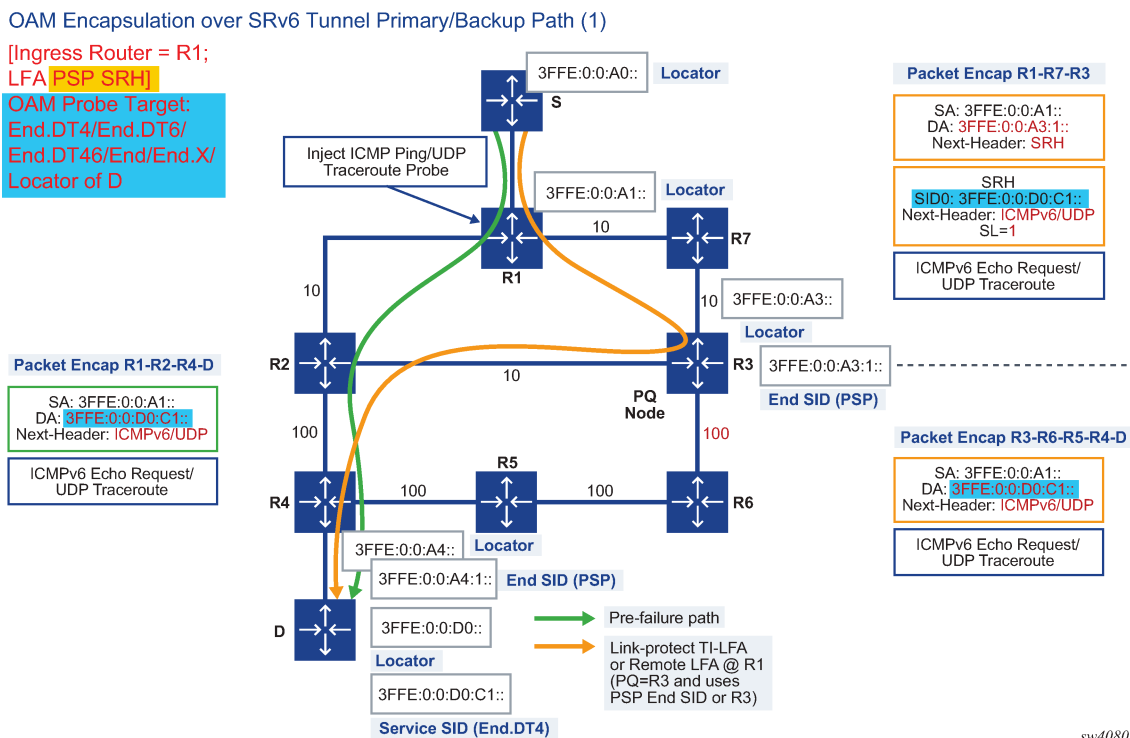
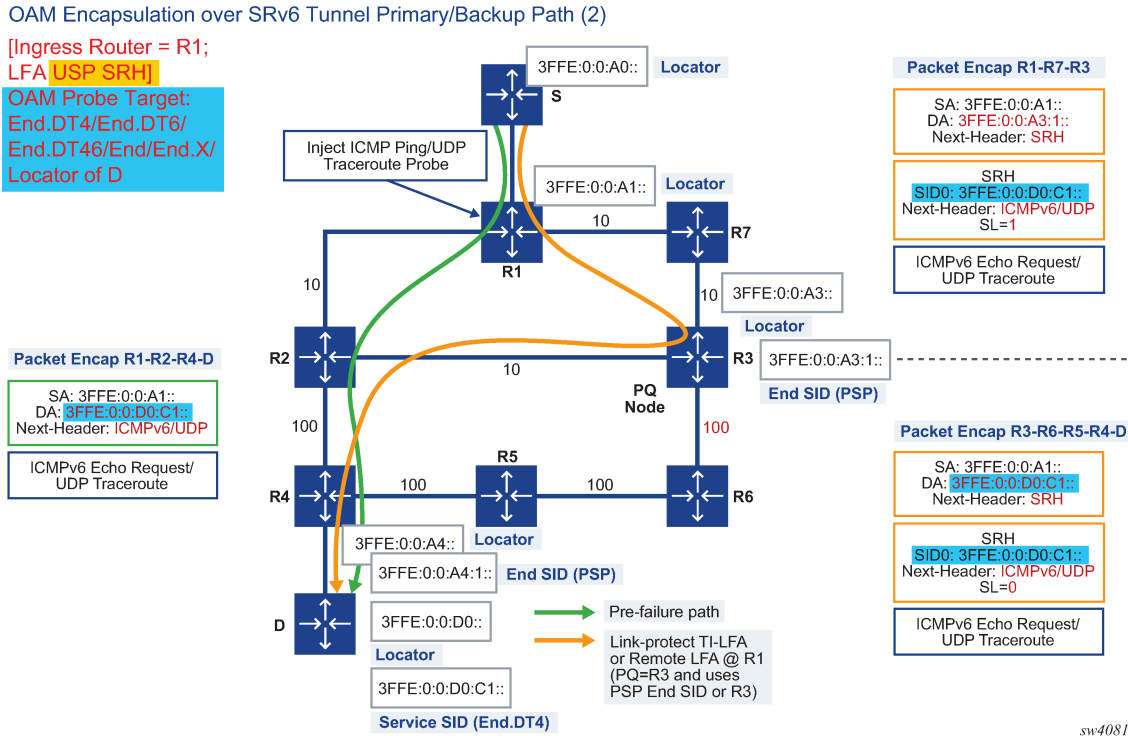


Figure 35: End-to-end packet encapsulation for ping or traceroute of a remote locator or SID (2)



### 3.1.9.1.4 Ping or traceroute of an IPv4/IPv6 VRF prefix resolved to an SRv6 shortest path or policy tunnel

This feature implements the existing behavior of a ping or a traceroute packet, originated at the ingress PE node, for a prefix resolved to an SRv6 tunnel. If the OAM ping or traceroute packet received from the CE router expires (with a hop-limit field value equal to or less than 1), the ingress PE node responds as per current behavior. If the packet does not expire (with a hop-limit field value greater than 1), it is forwarded over the SRv6 tunnel as a datapath packet.

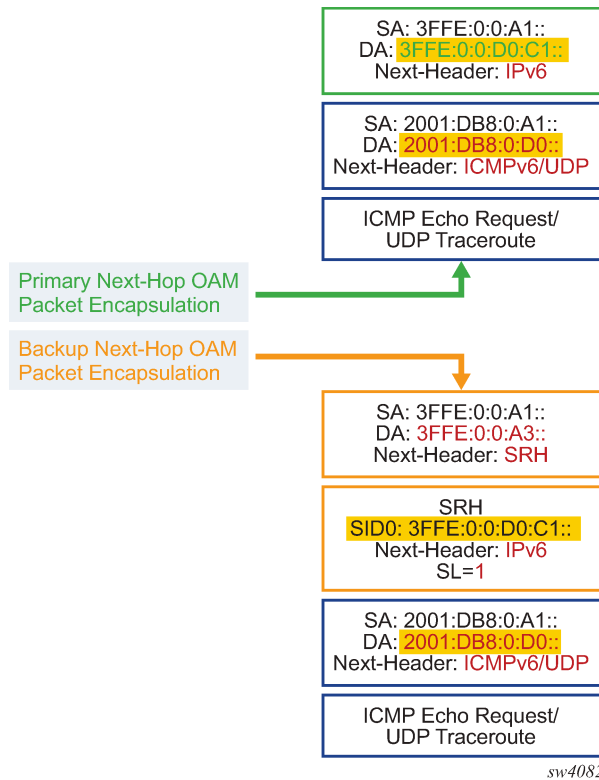
#### 3.1.9.1.4.1 Ingress PE router (sender node) behavior

The CPM-originated ping or traceroute packet is encoded with the:

- DA set to End.DT4, End.DT6, or End.DT46 SID
- next-header field set to IPv6 (IPv6 VRF prefix)
- next-header field of the inner IPv6 header is set to ICMPv6 (for ping) or to UDP (for traceroute)

The packet is encapsulated as shown in [Figure 36: Packet encapsulation for ping or traceroute of a VRF IPv6 prefix over an SRv6 tunnel](#). The LFA SRH is also shown in the packet encapsulation of the SRv6 tunnel to node D when the TI-LFA or remote LFA repair tunnel is activated.

Figure 36: Packet encapsulation for ping or traceroute of a VRF IPv6 prefix over an SRv6 tunnel



The outer IPv6 header hop-limit field is set to the default value 254.

### 3.1.9.1.4.2 Transit P router behavior

The packet is processed in the datapath, like any SRv6 user-data packet, by the transit router. See the *7750 SR and 7950 XRS Segment Routing and PCE User Guide* for further information.

### 3.1.9.1.4.3 Egress PE router behavior

At the target node, the packet is handed to the SRv6 Forwarding Path Extension (FPE) if the following apply:

- the packet matches a local locator prefix entry in the FIB
- the payload type is either IPv4 or IPv6

The egress datapath of the SRv6 FPE removes the SRv6 headers and passes the inner IPv6 packet to the ingress datapath, which performs the regular exception handling for a ping or a traceroute packet.

### 3.1.9.1.5 Ping or traceroute of an IPv4/IPv6 global routing instance prefix resolved to an SRv6 shortest path or policy tunnel

This behavior acts the same as described in [Ping or traceroute of an IPv4/IPv6 VRF prefix resolved to an SRv6 shortest path or policy tunnel](#).

### 3.1.9.1.6 Ping or Traceroute of an SRv6 policy

Use the ping tool to verify the connectivity of a specific static or BGP SRv6 policy and the traceroute tool to verify the path of the SRv6 policy. The objective of these tools is similar to LSP ping and LSP trace for an SR policy with an MPLS dataplane; they test that the SIDs specified in the SRv6 policy segment lists are programmed and reachable through the SRv6 policy and that the endpoint of the SRv6 policy is reachable through that policy. Both UDP and TCP transport are supported for traceroute.

From an encapsulation perspective, the SRv6 encapsulation is pushed onto the ping or traceroute packet and the next-header field in the Segment Routing Header (SRH) is set to ICMPv6, UDP or TCP (as applicable). Similar to seamless BFD (S-BFD), an additional address (the endpoint address of the SRv6 policy) is pushed as the final entry in the SRH to prevent SRH expiry in case the last SID in a segment list is a binding SID.

Use the following command to launch a ping for an SRv6 policy. You must configure the color and endpoint, and optionally configure the **segment-list** command option, to launch a ping.

```
ping srv6-policy color endpoint [segment-list]
```

Use the following command to launch a traceroute for an SRv6 policy. You must configure the command endpoint and color, and optionally configure the segment-list command option, to launch a traceroute.

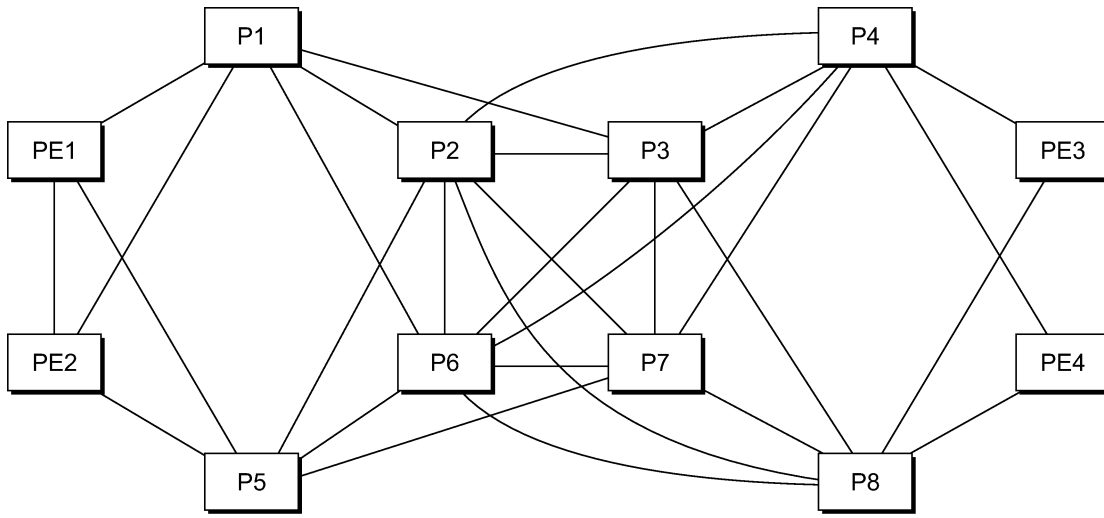
```
traceroute srv6-policy color endpoint [segment-list]
```

The router uses the tuple of **color**, **endpoint**, and optionally the **segment-list**, to match the SRv6 policy candidate path to send the ping or traceroute packet on the active candidate path. If the segment list ID is not specified, the router sends the ping or traceroute packet probe on the first available segment list that can be used for forwarding. For example, if a segment list is not specified the router sends on the lowest segment list that is UP, which means the top SID resolves or S-BFD is up on it. The test fails if the router does not find a matching active candidate path or a combination of a candidate path and segment list across all active static and BGP SR policies.

## 3.1.10 LDP tree trace: end-to-end testing of paths in an LDP ECMP network

[Figure 37: Network resilience using LDP ECMP](#) shows an IP/MPLS network which uses LDP ECMP for network resilience. Faults that are detected through IGP or LDP are corrected as soon as IGP and LDP re-converge. The impacted traffic is forwarded on the next available ECMP path as determined by the hash routine at the node that had a link failure.

Figure 37: Network resilience using LDP ECMP



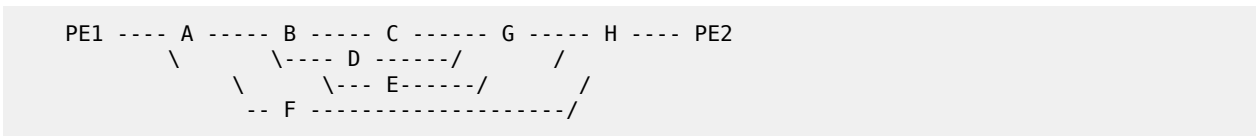
OSSG265

However, there are faults which the IGP/LDP control planes may not detect. These faults may be because of a corruption of the control plane state or of the data plane state in a node. Although these faults are very rare and mostly caused by misconfiguration, the LDP tree trace OAM feature is intended to detect these "silent" data plane and control plane faults. For example, it is possible that the forwarding plane of a node has a corrupt Next Hop Label Forwarding Entry (NHLFE) and keeps forwarding packets over an ECMP path only to have the downstream node discard them. This data plane fault can only be detected by an OAM tool that can test all possible end-to-end paths between the ingress LER and the egress LER. A corruption of the NHLFE entry can also result from a corruption in the control plane at that node.

### 3.1.11 LDP ECMP tree building

When the LDP tree trace feature is enabled, the ingress LER builds the ECMP tree for a specific FEC (egress LER) by sending LSP trace messages and including the LDP IPv4 Prefix FEC TLV as well as the downstream mapping TLV. To build the ECMP tree, the router LER inserts an IP address range drawn from the 127/8 space. When received by the downstream LSR, it uses this range to determine which ECMP path is exercised by any IP address or a sub-range of addresses within that range based on its internal hash routine. When the MPLS echo reply is received by the router LER, it records this information and proceed with the next echo request message targeted for a node downstream of the first LSR node along one of the ECMP paths. The sub-range of IP addresses indicated in the initial reply are used because the objective is to have the LSR downstream of the router LER pass this message to its downstream node along the first ECMP path.

The following figure illustrates the behavior through the following example adapted from RFC 8029:



LSR A has two downstream LSRs, B and F, for PE2 FEC. PE1 receives an echo reply from A with the Multipath Type set to 4, with low/high IP addresses of 127.1.1.1->127.1.1.255 for downstream LSR B and 127.2.1.1->127.2.1.255 for downstream LSR F. PE1 reflects this information to LSR B. B, which has three

downstream LSRs, C, D, and E, computes that 127.1.1.1->127.1.1.127 would go to C and 127.1.1.128->127.1.1.255 would go to D. B would then respond with 3 Downstream Mappings: to C, with Multipath Type 4 (127.1.1.1->127.1.1.127); to D, with Multipath Type 4 (127.1.1.127->127.1.1.255); and to E, with Multipath Type 0.

The router supports multipath type 0 and 8, and up to a maximum of 36 bytes for the multipath length and supports the LER part of the LDP ECMP tree building feature.

A user configurable parameter sets the frequency of running the tree trace capability. The minimum and default value is 60 minutes and the increment is 1 hour.

The router LER gets the list of FECs from the LDP FEC database. New FECs are added to the discovery list at the next tree trace and not when they are learned and added into the FEC database. The maximum number of FECs to be discovered with the tree building feature is limited to 500. The user can configure FECs to exclude the use of a policy profile.

### 3.1.12 Periodic path exercising

The periodic path exercising capability of the LDP tree trace feature runs in the background to test the LDP ECMP paths discovered by the tree building capability. The probe used is an LSP ping message with an IP address drawn from the sub-range of 127/8 addresses indicated by the output of the tree trace for this FEC.

The periodic LSP ping messages continuously probes an ECMP path at a user configurable rate of at least 1 message per minute. This is the minimum and default value. The increment is 1 minute. If an interface is down on a router LER, then LSP ping probes that normally go out this interface are not sent.

The LSP ping routine updates the content of the MPLS echo request message, specifically the IP address, as soon as the LDP ECMP tree trace has output the results of a new computation for the path in question.

### 3.1.13 LSP ping for RSVP P2MP LSP (P2MP)

The P2MP LSP ping complies to RFC 6425, *Detecting Data Plane Failures in Point-to-Multipoint Multiprotocol Label Switching (MPLS) - Extensions to LSP Ping*.

An LSP ping can be generated by entering the following OAM command:

```
- oam p2mp-lsp-ping lsp-name [p2mp-instance instance-name [s2l-dest-addr ip-address [...up to
5 max]]] [fc fc-name [profile {in | out}]] [size octets] [ttl label-ttl] [timeout timeout]
[detail]
```

The echo request message is sent on the active P2MP instance and is replicated in the data path over all branches of the P2MP LSP instance. By default, all egress LER nodes which are leaves of the P2MP LSP instance replies to the echo request message.

The user can reduce the scope of the echo reply messages by explicitly entering a list of addresses for the egress LER nodes that are required to reply. A maximum of 5 addresses can be specified in a single execution of the **p2mp-lsp-ping** command. If all 5 egress LER nodes are router nodes, they can parse the list of egress LER addresses and reply. RFC 6425 specifies that only the top address in the P2MP egress identifier TLV must be inspected by an egress LER. When interoperating with other implementations, the router egress LER responds if its address is anywhere in the list. Furthermore, if another vendor implementation is the egress LER, only the egress LER matching the top address in the TLV may respond.



If the user enters the same egress LER address more than once in a single `p2mp-lsp-ping` command, the head-end node displays a response to a single one and displays a single error warning message for the duplicate ones. When queried over SNMP, the head-end node issues a single response trap and issues no trap for the duplicates.

The **timeout** parameter should be set to the time it would take to get a response from all probed leaves under no failure conditions. For that purpose, its range extends to 120 seconds for a `p2mp-lsp-ping` from a 10 second `lsp-ping` for P2P LSP. The default value is 10 seconds.

The router head-end node displays a "Send\_Fail" error when a specific S2L path is down only if the user explicitly listed the address of the egress LER for this S2L in the **ping** command.

Similarly, the router head-end node displays the timeout error when no response is received for an S2L after the expiry of the timeout timer only if the user explicitly listed the address of the egress LER for this S2L in the **ping** command.

The user can configure a specific value of the **tll** parameter to force the echo request message to expire on a router branch node or a bud LSR node. The latter replies with a downstream mapping TLV for each branch of the P2MP LSP in the echo reply message. Note that a maximum of 16 downstream mapping TLVs can be included in a single echo reply message. It also sets the multipath type to zero in each downstream mapping TLV and not include any egress address information for the reachable egress LER nodes for this P2MP LSP.

If the router ingress LER node receives the new multipath type field with the list of egress LER addresses in an echo reply message from another vendor implementation, it ignores but not cause an error in processing the downstream mapping TLV.

If the ping expires at an LSR node which is performing a re-merge or cross-over operation in the data path between two or more ILMs of the same P2MP LSP, there is an echo reply message for each copy of the echo request message received by this node.

The output of the command without the **detail** parameter specified provides a high-level summary of error codes or success codes received.

The output of the command with the **detail** parameter specified shows a line for each replying node as in the output of the LSP ping for a P2P LSP.

The display is delayed until all responses are received or the timer configured in the timeout parameter expired. No other CLI commands can be entered while waiting for the display. A control-C (^C) command aborts the ping operation.

For more information about P2MP, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide*.

### 3.1.14 LSP trace for RSVP P2MP LSP

The P2MP LSP trace complies to RFC 6425. An LSP trace can be generated by entering the following OAM command:

```
- oam p2mp-lsp-trace lsp-name p2mp-instance instance-name s2l-dest-address ip-address [fc fc-name [profile {in | out}]] [size octets] [max-fail no-response-count] [probe-count probes-per-hop] [min-ttl min-label-ttl] [max-ttl max-label-ttl] [timeout timeout] [interval interval] [detail]
```

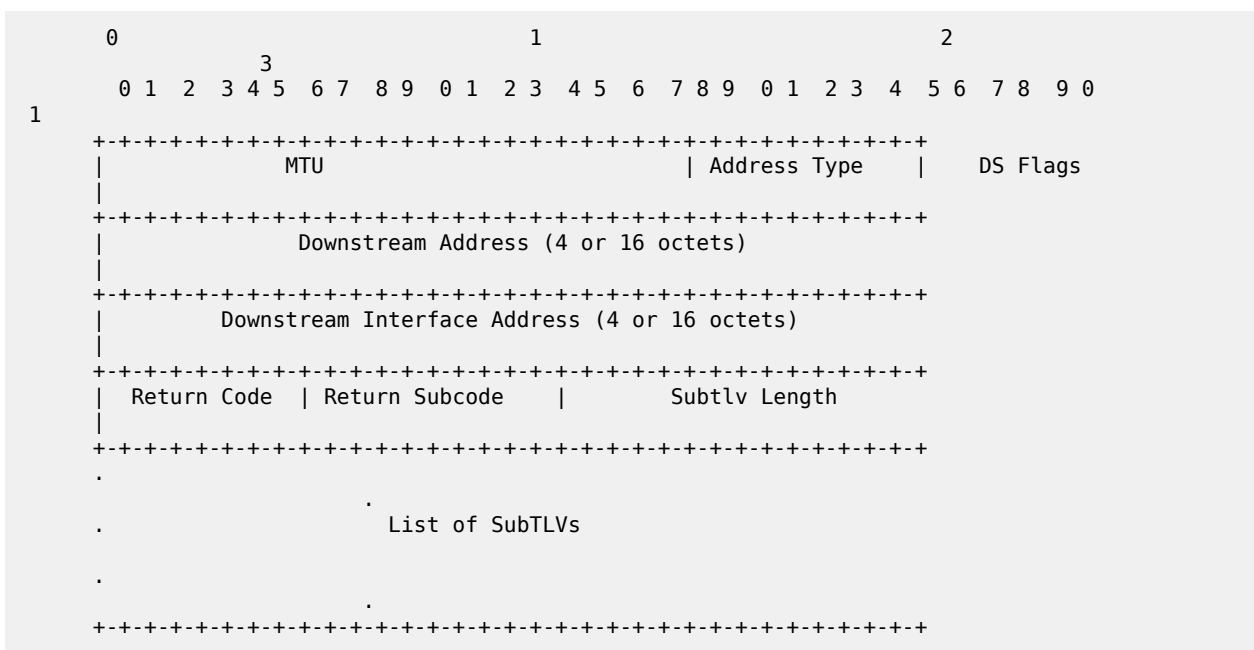
The LSP trace capability allows the user to trace the path of a single S2L path of a P2MP LSP. Its operation is similar to that of the **p2mp-lsp-ping** command but the sender of the echo reply request message includes the downstream mapping TLV to request the downstream branch information from a branch LSR or bud LSR. The branch LSR or bud LSR then also includes the downstream mapping TLV to

report the information about the downstream branches of the P2MP LSP. An egress LER does not include this TLV in the echo response message.

The **probe-count** parameter operates in the same way as in LSP trace on a P2P LSP. It represents the maximum number of probes sent per TTL value before giving up on receiving the echo reply message. If a response is received from the traced node before reaching maximum number of probes, then no more probes are sent for the same TTL. The sender of the echo request then increments the TTL and uses the information it received in the downstream mapping TLV to start sending probes to the node downstream of the last node which replied. This continues until the egress LER for the traced S2L path replied.

Because the command traces a single S2L path, the timeout and interval parameters keep the same value range as in LSP trace for a P2P LSP.

The P2MP LSP Trace makes use of the Downstream Detailed Mapping (DDMAP) TLV. The following excerpt from RFC 6424 details the format of the new DDMAP TLV entered in the path-destination belongs to one of the possible outgoing interface of the FEC.



The Downstream Detailed Mapping TLV format is derived from the Downstream Mapping (DSMAP) TLV format. The key change is that variable length and optional fields have been converted into sub-TLVs. The fields have the same use and meaning as in RFC 8029.

Similar to p2mp-lsp-ping, an LSP trace probe results on all egress LER nodes eventually receiving the echo request message but only the traced egress LER node replies to the last probe.

Also any branch LSR node or bud LSR node in the P2MP LSP tree may receive a copy of the echo request message with the TTL in the outer label expiring at this node. However, only a branch LSR or bud LSR which has a downstream branch over which the traced egress LER is reachable must respond.

When a branch LSR or BUD LSR node responds to the sender of the echo request message, it sets the global return code in the echo response message to RC=14 - "See DDMAP TLV for Return Code and Return Sub-Code" and the return code in the DDMAP TLV corresponding to the outgoing interface of the branch used by the traced S2L path to RC=8 - "Label switched at stack-depth <RSC>".

Because a single egress LER address, for example an S2L path, can be traced, the branch LSR or bud LSR node sets the multipath type of zero in the downstream mapping TLV in the echo response message as no egress LER address need to be included.

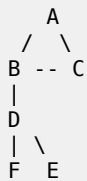
### 3.1.14.1 LSP trace behavior when S2L path traverses a re-merge node

When a 7450 ESS, 7750 SR or 7950 XRS LSR performs a re-merge of one or more ILMs of the P2MP LSP to which the traced S2L sub-LSP belongs, it may block the ILM over which the traced S2L resides. This causes the trace to either fail or to succeed with a missing hop.

The following is an example of this behavior.

S2L1 and S2L2 use ILMs which re-merge at node B. Depending on which ILM is blocked at B, the TTL=2 probe either yields two responses or times out.

```
S2L1 = ACBDF (to leaf F)
S2L2 = ABDE (to leaf E)
```



- **tracing S2L1 when ILM on interface C-B blocked at node B**

For TTL=1, A gets a response from C only as B does not have S2L1 on the ILM on interface A-B.

For TTL=2, assume A gets first the response from B which indicates a success. It then builds the next probe with TTL=3. B only passes the copy of the message arriving on interface A-B and drops the one arriving on interface C-B (treats it like a data packet because it does not expire at node B). This copy expires at F. However, F returns a "DSMappingMismatched" error message because the DDMAP TLV was the one provided by node B in TTL=2 step. The trace aborts at this point in time. However, A knows it received a second response from Node D for TTL=2 with a "DSMappingMismatched" error message.

If A gets the response from D first with the error code, it waits to see if it gets a response from B or it times out. In either case, it logs this status as **multiple replies received per probe** in the last probe history and aborts the trace.

- **tracing S2L2 when ILM on interface A-B blocked at node B**

For TTL=1, B responds with a success. C does not respond as it does not have an ILM for S2L2.

For TTL=2, B drops the copy coming on interface A-B. It receives a copy coming on interface B-C but drops it as the ILM does not contain S2L2. Node A times out. Next, node A generates a probe with TTL=3 without a DDMAP TLV. This time node D responds with a success and includes its downstream DDMAP TLV to node E. The rest of the path is discovered correctly. The traced path for S2L2 looks like: A-B-(\*)-D-E.

The router ingress LER detects a re-merge condition when it receives two or more replies to the same probe, such as the same TTL value. It displays the following message to the user regardless if the trace operation successfully reached the egress LER or was aborted earlier:

```
Probe returned multiple responses. Result may be inconsistent.
```

This warning message indicates to the user the potential of a re-merge scenario and that a p2mp-lsp-ping command for this S2L should be used to verify that the S2L path is not defective.

The router ingress LER behavior is to always proceed to the next ttl probe when it receives an OK response to a probe or when it times out on a probe. If however it receives replies with an error return code, it must wait until it receives an OK response or it times out. If it times out without receiving an OK reply, the LSP trace must be aborted.

Possible echo reply messages and corresponding ingress LER behaviors are described in [Table 8: Echo reply messages and ingress LER behavior](#).

*Table 8: Echo reply messages and ingress LER behavior*

Echo reply message	Ingress LER behavior
One or more error return codes + OK	Display OK return code. Proceed to next ttl probe. Display warning message at end of trace.
OK + one or more error return codes	Display OK return code. Proceed to next ttl probe right after receiving the OK reply but keep state that more replies received. Display warning message at end of trace.
OK + OK	Should not happen for re-merge but would continue trace on 1st OK reply. This is the case when one of the branches of the P2MP LSP is activating the P2P bypass LSP. In this case, the head-end node receives a reply from both a regular P2MP LSR which has the ILM for the traced S2L and from an LSR switching the P2P bypass for other S2Ls. The latter does not have context for the P2MP LSP being tunneled but responds after doing a label stack validation.
One error return code + timeout	Abort LSP trace and display error code. Ingress LER cannot tell the error occurred of a re-merge condition.
More than one error return code + timeout	Abort LSP trace and display first error code. Display warning message at end of trace.
Timeout on probe without any reply	Display "*" and proceed to next ttl probe.

### 3.1.15 Tunneling of ICMP reply packets over MPLS LSP

This feature enables the tunneling of ICMP reply packets over MPLS LSP at an LSR node as per RFC 3032. At an LSR node, including an ABR, ASBR, or data path Router Reflector (RR) node, the user enables the ICMP tunneling feature globally on the system using the **config>router>icmp-tunneling** command.

This feature supports tunneling ICMP replies to a UDP traceroute message. It does not support tunneling replies to an icmp ping message. The LSR part of this feature consists of crafting the reply ICMP packet of type=11- 'time exceeded', with a source address set to a local address of the LSR node, and appending the

IP header and leading payload octets of the original datagram. The system skips the lookup of the source address of the sender of the label TTL expiry packet, which becomes the destination address of the ICMP reply packet. Instead, CPM injects the ICMP reply packet in the forward direction of the MPLS LSP the label TTL expiry packet was received from. The TTL of pushed labels should be set to 255.

The source address of the ICMP reply packet is determined as follows:

- The LSR uses the address of the outgoing interface for the MPLS LSP. Note that with LDP LSP or BGP LSP, multiple ECMP next-hops can exist in which case the first outgoing interface is selected.
- If the interface does not have an address of the same family (IPv4 or IPv6) as the ICMP packet, then the system address of the same family is selected. If one is not configured, the packet is dropped.

When the packet is received by the egress LER, it performs a regular user packet lookup in the data path in the GRT context for BGP shortcut, 6PE, and BGP label route prefixes, or in VPRN context for VPRN and 6VPE prefixes. It then forwards it to the destination, which is the sender of the original packet which TTL expired at the LSR.

If the egress LER does not have a route to the destination of the ICMP packet, it drops the packets.

The rate of the tunneled ICMP replies at the LSR can be directly or indirectly controlled by the existing IOM level and CPM levels mechanisms. Specifically, the rate of the incoming UDP traceroute packets received with a label stack can be controlled at ingress IOM using the distributed CPU protection feature. The rate of the ICMP replies by CPM can also be directly controlled by configuring a system wide rate limit for packets ICMP replies to MPLS expired packets which are successfully forwarded to CPM using the command 'configure system security vprn-network-exceptions'. Note that while this command's name refers to VPRN service, this feature rate limits ICMP replies for packets received with any label stack, including VPRN and shortcuts.

The 7450 ESS, 7750 SR and 7950 XRS router implementation supports appending to the ICMP reply of type Time Exceeded the MPLS label stack object defined in RFC 4950. It does not include it in the ICMP reply type of Destination unreachable.

The new MPLS Label Stack object allows an LSR to include label stack information including label value, EXP, and TTL field values, from the encapsulation header of the packet that expired at the LSR node. The ICMP message continues to include the IP header and leading payload octets of the original datagram.

To include the MPLS Label Stack object, the SR OS implementation adds support of RFC 4884, *Extended ICMP to Support Multi-Part Messages*, which defines extensions for a multi-part ICMPv4/v6 message of type Time Exceeded. Section 5 of RFC 4884 defines backward compatibility of the new ICMP message with extension header with prior standard and proprietary extension headers.

To guarantee interoperability with third party implementations deployed in customer networks, the router implementation is able to parse in the receive side all possible encapsulations formats as defined in Section 5 of RFC 4884. Specifically:

The new MPLS Label Stack object allow an LSR to include label stack information including label value, EXP, and TTL field values, from the encapsulation header of the packet that expired at the LSR node. The ICMP message continues to include the IP header and leading payload octets of the original datagram.

- If the length attribute is zero, it is treated as a compliant message and the router implementation processes the original datagram field of size equal to 128 bytes and with no extension header.
- If the length attribute is not included, it is treated as a non-compliant message and the router implementation processes the original datagram field of size equal to 128 bytes and also look for a valid extension header following the 128 byte original datagram field. If the extension is valid, it is processed accordingly, if not it is assumed the remainder of the packet is still part of the original datagram field and process it accordingly. Note that the router implementation only validates the ICMP extension version

number and not the checksum field in the extension header. The checksum of the main time exceeded message is also not validated as per prior implementation.

- An ICMP reply message is dropped if it includes more than one MPLS label object. In general when a packet is dropped because of an error in the packet header or structure, the traceroute timeouts and an error message is not displayed.
- When processing the received ICMP reply packet, an unsupported extension header is skipped.

In the transmit side, when the MPLS Label Stack object is added as an extension to the ICMP reply message, it is appended to the message immediately following the "original datagram" field taken from the payload of the received traceroute packet. The size of the appended "original datagram" field contains exactly 128 octets. If the original datagram did not contain 128 octets, the "original datagram" field is zero padded to 128 octets.

For example output of the traceroute OAM tool when the ICMP tunneling feature is enabled see, [Traceroute with ICMP tunneling in common applications](#).

### 3.1.16 QoS handling of tunneled ICMP reply packets

When the ICMP reply packet is generated in CPM, its FC is set by default to NC1 with the corresponding default ToS byte value of 0xC0. The DSCP value can be changed by configuring a different value for an ICMP application under the **config>router>sgt-qos icmp** context.

When the packet is forwarded to the outgoing interface, the packet is queued in the egress network queue corresponding to its CPM assigned FC and profile parameter values. The marking of the packet's EXP is dictated by the {FC, profile}-to-EXP mapping in the network QoS policy configured on the outgoing network interface. The ToS byte, and DSCP value for that matter, assigned by CPM are not modified by the IOM.

### 3.1.17 Summary of UDP traceroute behavior with and without ICMP tunneling

At a high level, the major difference in the behavior of the UDP traceroute when ICMP tunneling is enabled at an LSR node is that the LSR node tunnels the ICMP reply packet toward the egress of the LSP without looking up the traceroute sender's address. When ICMP tunneling is disabled, the LSR looks it up and replies if the sender is reachable. However there are additional differences in the two behaviors and they are summarized in the following.

- **icmp-tunneling disabled/IPv4 LSP/IPv4 traceroute**

Ingress LER, egress LER, and LSR attempt to reply to the UDP traceroute of both IPv4 and VPN-IPv4 routes.

For VPN-IPv4 routes, the LSR attempts to reply but it may not find a route and in such a case the sender node times out. In addition, the ingress and egress ASBR nodes in VPRN inter-AS option B do not respond as in current implementation and the sender times out.

- **icmp-tunneling disabled/IPv4 LSP/IPv6 traceroute**

Ingress LER and egress LER reply to traceroute of both IPv6 and VPN-IPv6 routes. LSR does not reply.

- **icmp-tunneling enabled/IPv4 LSP/IPv4 traceroute**

Ingress LER and egress LER reply directly to the UDP traceroute of both IPv4 and VPN-IPv4 routes. LSR tunnels the reply to endpoint of the LSP to be forwarded from there to the source of the traceroute.

For VPN-IPv4 routes, the ingress and egress ASBR nodes in VPRN inter-AS option B also tunnels the reply to the endpoint of the LSP and therefore there is no timeout at the sender node like in the case when icmp-tunneling is disabled.

- **icmp-tunneling enabled/IPv4 LSP/IPv6 traceroute**

Ingress LER and egress LER reply directly to the UDP traceroute of both IPv6 and VPN-IPv6 routes. LSR tunnels the reply to endpoint of the LSP to be forwarded from there to the source of the traceroute.

For VPN-IPv6 routes, the ingress and egress ASBR nodes in VPRN inter-AS option B also tunnels the reply to the endpoint of the LSP like in the case when icmp-tunneling is disabled.

In the presence of ECMP, CPM generated UDP traceroute packets are not sprayed over multiple ECMP next-hops. The first outgoing interface is selected. In addition, a LSR ICMP reply to a UDP traceroute is also forwarded over the first outgoing interface regardless if ICMP tunneling is enabled or not. When ICMP tunneling is enabled, it means the packet is tunneled over the first downstream interface for the LSP when multiple next-hops exist (LDP FEC or BGP label route). In all cases, the ICMP reply packet uses the outgoing interface address as the source address of the reply packet.

### 3.1.18 SDP diagnostics

The router SDP diagnostics are SDP ping and SDP MTU path discovery.

### 3.1.19 SDP ping

SDP ping performs in-band unidirectional or round-trip connectivity tests on SDPs. The SDP ping OAM packets are sent in-band, in the tunnel encapsulation, so it follows the same path as traffic within the service. The SDP ping response can be received out-of-band in the control plane, or in-band using the data plane for a round-trip test.

For a unidirectional test, SDP ping tests:

- egress SDP ID encapsulation
- ability to reach the far-end IP address of the SDP ID within the SDP encapsulation
- path MTU to the far-end IP address over the SDP ID
- forwarding class mapping between the near-end SDP ID encapsulation and the far-end tunnel termination

For a round-trip test, SDP ping uses a local egress SDP ID and an expected remote SDP ID. Because SDPs are unidirectional tunnels, the remote SDP ID must be specified and must exist as a configured SDP ID on the far-end router SDP round trip testing is an extension of SDP connectivity testing with the additional ability to test:

- remote SDP ID encapsulation
- potential service round trip time
- round trip path MTU
- round trip forwarding class mapping

### 3.1.20 SDP MTU path discovery

In a large network, network devices can support a variety of packet sizes that are transmitted across its interfaces. This capability is referred to as the Maximum Transmission Unit (MTU) of network interfaces. It is important to understand the MTU of the entire path end-to-end when provisioning services, especially for virtual leased line (VLL) services where the service must support the ability to transmit the largest customer packet.

The Path MTU discovery tool provides a powerful tool that enables service provider to get the exact MTU supported by the network's physical links between the service ingress and service termination points (accurate to one byte).

### 3.1.21 Service diagnostics

Nokia's Service ping feature provides end-to-end connectivity testing for an individual service. Service ping operates at a higher level than the SDP diagnostics in that it verifies an individual service and not the collection of services carried within an SDP.

Service ping is initiated from a router to verify round-trip connectivity and delay to the far-end of the service. Nokia's implementation functions for both GRE and MPLS tunnels and tests the following from edge-to-edge:

- tunnel connectivity
- VC label mapping verification
- service existence
- service provisioned parameter verification
- round trip path verification
- service dynamic configuration verification

### 3.1.22 VPLS MAC diagnostics

While the LSP ping, SDP ping and service ping tools enable transport tunnel testing and verify whether the correct transport tunnel is used, they do not provide the means to test the learning and forwarding functions on a per-VPLS-service basis.

It is conceivable, that while tunnels are operational and correctly bound to a service, an incorrect Forwarding Database (FDB) table for a service could cause connectivity issues in the service and not be detected by the ping tools. Nokia has developed VPLS OAM functionality to specifically test all the critical functions on a per-service basis. These tools are based primarily on the IETF document *draft-stokes-vkompella-ppvpn-hvpls-oam-xx.txt*, *Testing Hierarchical Virtual Private LAN Services*.

The VPLS OAM tools are:

- **MAC ping** provides an end-to-end test to identify the egress customer-facing port where a customer MAC was learned. MAC ping can also be used with a broadcast MAC address to identify all egress points of a service for the specified broadcast MAC.
- **MAC trace** provides the ability to trace a specified MAC address hop-by-hop until the last node in the service domain. An SAA test with MAC trace is considered successful when there is a reply from a far-end node indicating that they have the destination MAC address on an egress SAP or the CPM.



- **CPE ping** provides the ability to check network connectivity to the specified client device within the VPLS. CPE ping returns the MAC address of the client, as well as the SAP and PE at which it was learned.
- **MAC populate** allows specified MAC addresses to be injected in the VPLS service domain. This triggers learning of the injected MAC address by all participating nodes in the service. This tool is generally followed by MAC ping or MAC trace to verify if correct learning occurred.
- **MAC purge** allows MAC addresses to be flushed from all nodes in a service domain.

### 3.1.23 MAC ping

For a MAC ping test, the destination MAC address (unicast or multicast) to be tested must be specified. A MAC ping packet is sent through the data plane. The ping packet goes out with the data plane format.

In the data plane, a MAC ping is sent with a VC label TTL of 255. This packet traverses each hop using forwarding plane information for next hop, VC label, and so on. The VC label is swapped at each service-aware hop, and the VC TTL is decremented. If the VC TTL is decremented to 0, the packet is passed up to the management plane for processing. If the packet reaches an egress node, and would be forwarded out a customer facing port, it is identified by the OAM label below the VC label and passed to the management plane.

MAC pings are flooded when they are unknown at an intermediate node. They are responded to only by the egress nodes that have mappings for that MAC address.

### 3.1.24 MAC trace

A MAC trace functions like an LSP trace with some variations. Operations in a MAC trace are triggered when the VC TTL is decremented to 0.

Like a MAC ping, a MAC trace is sent using the data plane.

When a traceroute request is sent via the data plane, the data plane format is used. The reply can be via the data plane or the control plane.

A data plane MAC traceroute request includes the tunnel encapsulation, the VC label, and the OAM, followed by an Ethernet DLC, a UDP, and IP header. If the mapping for the MAC address is known at the sender, the data plane request is sent down the known SDP with the appropriate tunnel encapsulation and VC label. If the mapping is not known, it is sent down every SDP (with the appropriate tunnel encapsulation per SDP and appropriate egress VC label per SDP binding).

The tunnel encapsulation TTL is set to 255. The VC label TTL is initially set to the min-ttl (default is 1). The OAM label TTL is set to 2. The destination IP address is the all-routers multicast address. The source IP address is the system IP of the sender.

The destination UDP port is the LSP ping port. The source UDP port is whatever the system provides (this source UDP port is the demultiplexer that identifies the particular instance that sent the request, when correlating the reply).

The Reply Mode is either 3 (that is, reply using the control plane) or 4 (that is, reply through the data plane), depending on the reply-control option. By default, the data plane request is sent with Reply Mode 3 (control plane reply).

The Ethernet DLC header source MAC address is set to either the system MAC address (if no source MAC is specified) or to the specified source MAC. The destination MAC address is set to the specified destination MAC. The EtherType is set to IP.

### 3.1.25 CPE ping

The Nokia-specific CPE ping function provides a common approach to determine if a destination IPv4 address can be resolved to a MAC address beyond the Layer 2 PE, in the direction of the CPE. The function is supported for both VPLS and Epipe services and on a number of different connection types. The service type determines the packet format for network connection transmissions. The transmission of the packet from a PE egressing an access connection is a standard ARP packet. This allows for next-hop resolution for even unmanaged service elements. In many cases, responses to ICMP echo requests are restricted to trusted network segments only; however, ARP packets are typically processed.

If the ARP response is processed on a local SAP connection on the same node from which the command was executed, the detailed SAP information is returned as part of the display function. If the response is not local, the format of the display depends on the service type.

The VPLS service construct is multipoint by nature, and simply returning a positive response to a reachability request would not supply enough information. For this reason, VPLS service CPE ping requests use the Nokia-specific MAC ping packet format. Execution of the CPE ping command generates a MAC ping packet using a broadcast Layer 2 address on all non-access ports. This packet allows for more information about the location of the target. A positive result displays the IP address of the Layer 2 PE and SAP information for the target location.

Each PE, including the local PE, that receives a MAC ping proxies an ARP request on behalf of the original source, as part of the CPE ping function. If a response is received for the ARP request, the Layer 2 PE processes the request, translates the ARP response, and responds back to the initial source with the appropriate MAC ping response and fields.

The MAC ping OAM tool makes it possible to detect whether a particular IPv4 address and MAC address have been learned in a VPLS, and on which SAP the target was found.

The Epipe service construction is that of cross-connection, and returning a positive response to a reachability request is an acceptable approach. For this reason, Epipe service CPE ping requests use standard ARP requests and proxy ARP processing. A positive result displays **remote-SAP** for any non-local responses. Because Epipe services are point-to-point, the path toward the remote SAP for the service should already be understood.

Nokia recommends that a source IP address of all zeros (0.0.0.0) is used, which prevents the exposure of the provider IP address to the CPE.

The CPE ping function requires symmetrical data paths for correct functionality. Issues may arise when the request egresses a PE and the response arrives on a related but different PE. When dealing with asymmetrical paths, the **return-control** option may be used to bypass some of the asymmetrical path issues. Asymmetrical paths can be common in all active multi-homing solutions.

For all applications except basic VPLS services (SAP and SDP bindings without a PBB context), CPE ping functionality requires minimum FP2-based hardware for all connections that may be involved in the transmission or processing of the proxy function.

This approach should only be considered for unmanaged solutions where standard Ethernet CFM (ETH-CFM) functions cannot be deployed. ETH-CFM has a robust set of fault and performance functions that are purpose-built for Ethernet services and transport.

Connection types used to support VPLS and Epipes include SAPs, SDP bindings, B-VPLS, BGP-AD, BGP-VPWS, BGP-VPLS, and MPLS-EVPN.

### 3.1.26 CPE ping for PBB Epipe

CPE ping has been supported for VPLS services since Release 3.0 of SR OS. It enables the connectivity of the access circuit between a VPLS PE and a CPE to be tested, even if the CPE is unmanaged and, therefore, the service provider cannot run standardized Ethernet OAM to the CPE. The command **cpe-ping** for a specific destination IP address within a VPLS is translated into a **mac-ping** toward a broadcast MAC address. All destinations within the VPLS context are reached by this ping to the broadcast MAC address. At all these destinations, an ARP is triggered for the specific IP address (with the IP destination address equal to the address from the request, mac-da equal to all ones, mac-sa equal to the CPM-mac-address and the IP source address, which is the address found in the request). The destination receiving a response replies back to the requester.

Release 10.0 extended the CPE ping command for local, distributed, and PBB Epipe services provisioned over a PBB VPLS. CPE ping for Epipe implements an alternative behavior to CPE ping for VPLS that enables fate sharing of the CPE ping request with the Epipe service. Any PE within the Epipe service (the source PE) can launch the CPE ping. The source PE builds an ARP request and encapsulates it to be sent in the Epipe as if it came from a customer device by using its chassis MAC as the source MAC address. The ARP request then egresses the remote PE device as any other packets on the Epipe. The remote CPE device responds to the ARP and the reply is transparently sent on the Epipe toward the source PE. The source PE then looks for a match on its chassis MAC in the inner customer DA. If a match is found, the source PE device intercepts this response packet.

This method is supported regardless of whether the network uses SDPs or SAPs. It is configured using the existing **oam>cpe-ping** CLI command.



**Note:** This feature does not support IPv6 CPEs.

#### 3.1.26.1 Hardware support

This feature supports FP2 and later and applies only to the 7450 ESS and 7750 SR.

To launch **cpe-ping** on an Epipe, all of the following must be true:

- All SAPs on the Epipe must be provisioned on slots that are chassis mode D compatible.
- If bound to a PBB tunnel, all SAPs on the B-VPLS must be provisioned on slots that are chassis mode D compatible.
- If the Epipe or the B-VPLS (in the case of PBB Epipe) uses SDP-bindings, the system configuration must be network chassis mode D compatible.

### 3.1.27 MAC populate

MAC populate is used to send a message through the flooding domain to learn a MAC address as if a customer packet with that source MAC address had flooded the domain from that ingress point in the service. This allows the provider to craft a learning history and engineer packets in a particular way to test forwarding plane correctness.

The MAC populate request is sent with a VC TTL of 1, which means that it is received at the forwarding plane at the first hop and passed directly up to the management plane. The packet is then responded to by populating the MAC address in the forwarding plane, similar to a conventional learn, although the MAC is an OAM-type MAC in the FDB to distinguish it from customer MAC addresses.

This packet is then taken by the control plane and flooded out the flooding domain (squelching, appropriately, the sender and other paths that would be squelched in a typical flood).

This controlled population of the FDB is very important to manage the expected results of an OAM test. The same functions are available by sending the OAM packet as a UDP/IP OAM packet. It is then forwarded to each hop and the management plane has to do the flooding.

Options for MAC populate are to force the MAC in the table to type OAM (in case it already existed as dynamic or static or an OAM-induced learning with some other binding). This prevents new dynamic learning from overwriting the existing OAM MAC entry, to allow customer packets with this MAC to either ingress or egress the network, while still using the OAM MAC entry.

Finally, an option to flood the MAC populate request causes each upstream node to learn the MAC, populate the local FDB with an OAM MAC entry, and to flood the request along the data plane using the flooding domain.

An age can be provided to age a particular OAM MAC after a different interval than other MACs in a FDB.

### 3.1.28 MAC purge

MAC purge is used to clear the FDBs of any learned information for a particular MAC address. This allows one to do a controlled OAM test without learning induced by customer packets. In addition to clearing the FDB of a particular MAC address, the purge can also indicate to the control plane not to allow further learning from customer packets. This allows the FDB to be clean, and be populated only via a MAC Populate.

MAC purge follows the same flooding mechanism as the MAC populate.

### 3.1.29 VLL diagnostics

### 3.1.30 VCCV ping

VCCV ping is used to check connectivity of a VLL in-band. It checks that the destination (target) PE is the egress for the Layer 2 FEC. It provides a cross-check between the data plane and the control plane. It is in-band, meaning that the VCCV ping message is sent using the same encapsulation and along the same path as user packets in that VLL. This is equivalent to the LSP ping for a VLL service. VCCV ping reuses an LSP ping message format and can be used to test a VLL configured over an MPLS and GRE SDP.

#### 3.1.30.1 VCCV-ping application

VCCV effectively creates an IP control channel within the pseudowire between PE1 and PE2. PE2 should be able to distinguish on the receive side VCCV control messages from user packets on that VLL. There are three possible methods of encapsulating a VCCV message in a VLL which translates into three types of control channels:

- Use of a Router Alert Label immediately above the VC label. This method has the drawback that if ECMP is applied to the outer LSP label (for example, transport label), the VCCV message does not follow the same path as the user packets. This effectively means it does not troubleshoot the appropriate path. This method is supported by the 7450 ESS, 7750 SR, and 7950 XRS routers.

- Use of the OAM control word as shown:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|0 0 0 1| FmtID |   Reserved   |           Channel Type           |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The first nibble is set to 0x1. The Format ID and the reserved fields are set to 0 and the channel type is the code point associated with the VCCV IP control channel as specified in the PWE3 IANA registry (RFC 4446). The channel type value of 0x21 indicates that the Associated Channel carries an IPv4 packet.

The use of the OAM control word assumes that the draft-martini control word is also used on the user packets. This means that if the control word is optional for a VLL and is not configured, the PE node only advertises the router alert label as the CC capability in the Label Mapping message. This method is supported by the 7450 ESS, 7750 SR and 7950 XRS routers.

- Set the TTL in the VC label to 1 to force PE2 control plane to process the VCCV message. This method is not guaranteed to work under all circumstances. For instance, the draft mentions some implementations of penultimate hop popping overwrite the TTL field. This method is not supported by the 7450 ESS, 7750 SR, and 7950 XRS routers.

When sending the label mapping message for the VLL, PE1 and PE2 must indicate which of the above OAM packet encapsulation methods (for example, which control channel type) they support. This is accomplished by including an optional VCCV TLV in the pseudowire FEC Interface Parameter field. The format of the VCCV TLV is shown below:

```

0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|           0x0c           |           0x04           | CC Types | CV Types |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Note that the absence of the optional VCCV TLV in the Interface parameters field of the pseudowire FEC indicates the PE has no VCCV capability.

The Control Channel (CC) Type field is a bitmask used to indicate if the PE supports none, one, or many control channel types.

- 0x00 None of the following VCCV control channel types are supported
- 0x01 PWE3 OAM control word
- 0x02 MPLS Router Alert Label
- 0x04 MPLS inner label TTL = 1

If both PE nodes support more than one of the CC types, then the router PE uses of the one with the lowest type value. For instance, OAM control word is used in preference to the MPLS router alert label.

The Connectivity Verification (CV) bitmask field is used to indicate the specific type of VCCV packets to be sent over the VCCV control channel. The valid values are:

0x00 None of the below VCCV packet type are supported.

0x01 icmp ping. Not applicable to a VLL over a MPLS or GRE SDP and is therefore not supported by the 7450 ESS, 7750 SR, and 7950 XRS routers.

0x02 LSP ping. This is used in VCCV ping application and applies to a VLL over an MPLS or a GRE SDP. This is supported by the 7450 ESS, 7750 SR, and 7950 XRS routers.

A VCCV ping is an LSP echo request message as defined in RFC 8029. It contains an L2 FEC stack TLV which must include within the sub-TLV type 10 "FEC 128 Pseudowire". It also contains a field which indicates to the destination PE which reply mode to use. There are four reply modes defined in RFC 8029:

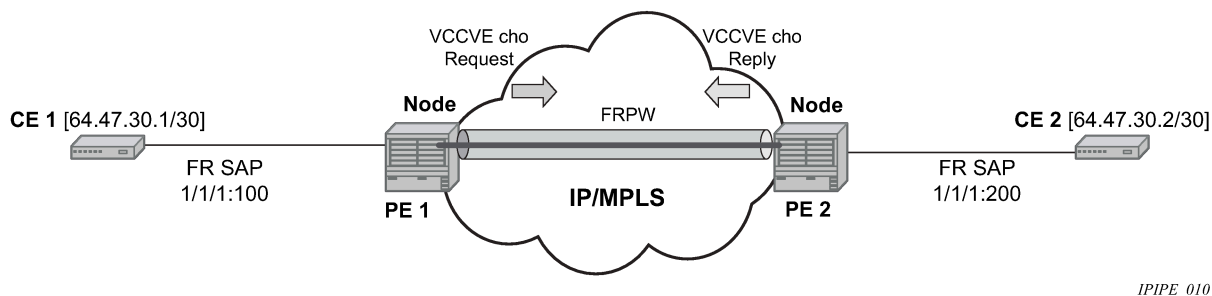
Reply mode, meaning:

- Do not reply. This mode is supported by the routers.
- Reply via an IPv4/IPv6 UDP packet. This mode is supported by the routers.
- Reply with an IPv4/IPv6 UDP packet with a router alert. This mode sets the router alert bit in the IP header and is not be confused with the CC type which makes use of the router alert label. This mode is not supported by the routers.
- Reply via application level control channel. This mode sends the reply message inband over the pseudowire from PE2 to PE1. PE2 encapsulates the Echo Reply message using the CC type negotiated with PE1. This mode is supported by the routers.

The reply is an LSP echo reply message as defined in RFC 8029. The message is sent as per the reply mode requested by PE1. The return codes supported are the same as those supported in the router LSP ping capability.

The VCCV ping feature is in addition to the service ping OAM feature which can be used to test a service between router nodes. The VCCV ping feature can test connectivity of a VLL with any third party node which is compliant to RFC 5085.

Figure 38: VCCV-ping application



### 3.1.30.2 VCCV ping in a multi-segment pseudowire

Figure 39: VCCV ping over a multi-segment pseudowire shows an example of an application of VCCV ping over a multi-segment pseudowire.

Pseudowire switching is a method for scaling a large network of VLL or VPLS services by removing the need for a full mesh of T-LDP sessions between the PE nodes as the number of these nodes grow over time. Pseudowire switching is also used whenever there is a need to deploy a VLL service across two separate routing domains.

In the network, a Termination PE (T-PE) is where the pseudowire originates and terminates. The Switching PE (S-PE) is the node which performs pseudowire switching by cross-connecting two spoke SDPs.

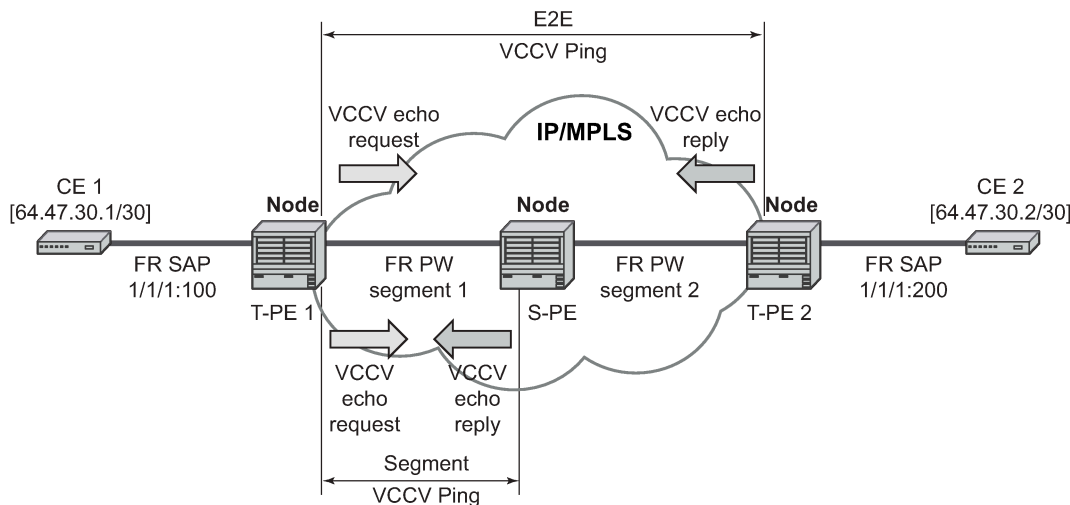
VCCV ping is able to ping to a destination PE. A VLL FEC ping is a message sent by T-PE1 to test the FEC at T-PE2. The operation at T-PE1 and T-PE2 is the same as in the case of a single-segment

pseudowire. The pseudowire switching node, S-PE1, pops the outer label, swaps the inner (VC) label, decrements the TTL of the VC label, and pushes a new outer label. The PE1 node does not process the VCCV OAM Control Word unless the VC label TTL expires. In that case, the message is sent to the CPM for further validation and processing. This is the method described in draft-hart-pwe3-segmented-pw-vcv.

Note that the originator of the VCCV ping message does not need to be a T-PE node; it can be an S-PE node. The destination of the VCCV ping message can also be an S-PE node.

VCCV trace to trace the entire path of a pseudowire with a single command issued at the T-PE. This is equivalent to LSP trace and is an iterative process by which T-PE1 sends successive VCCV ping messages while incrementing the TTL value, starting from TTL=1. The procedure for each iteration is the same as above and each node in which the VC label TTL expires checks the FEC and replies with the FEC to the downstream S-PE or T-PE node. The process is terminated when the reply is from T-PE2 or when a timeout occurs.

Figure 39: VCCV ping over a multi-segment pseudowire



OSSG113

### 3.1.31 Automated VCCV-trace capability for MS-pseudowire

Although tracing of the MS-pseudowire path is possible using the methods described in previous sections, these require multiple manual iterations and requires that the FEC of the last pseudowire segment to the target T-PE/S-PE is known as priori at the node originating the echo request message for each iteration. This mode of operation is referred to as a "ping" mode.

The automated VCCV-trace can trace the entire path of a pseudowire with a single command issued at the T-PE or at an S-PE. This is equivalent to LSP-trace and is an iterative process by which the ingress T-PE or T-PE sends successive VCCV-ping messages with incrementing the TTL value, starting from TTL=1.

The method is described in draft-hart-pwe3-segmented-pw-vcv, *VCCV Extensions for Segmented Pseudo-Wire*, and is pending acceptance by the PWE3 working group. In each iteration, the source T-PE or S-PE builds the MPLS echo request message in a way similar to [VCCV ping](#). The first message with TTL=1 has the next-hop S-PE T-LDP session source address in the Remote PE Address field in the pseudowire FEC TLV. Each S-PE which terminates and processes the message includes in the MPLS echo reply message the FEC 128 TLV corresponding the pseudowire segment to its downstream node. The inclusion of the FEC TLV in the echo reply message is allowed in RFC 8029. The source T-PE or S-PE

can then build the next echo reply message with TTL=2 to test the next-next hop for the MS-pseudowire. It copies the FEC TLV it received in the echo reply message into the new echo request message. The process is terminated when the reply is from the egress T-PE or when a timeout occurs. If specified, the max-ttl parameter in the vccv-trace command stops on S-PE before reaching T-PE.

The results VCCV-trace can be displayed for a fewer number of pseudowire segments of the end-to-end MS-pseudowire path. In this case, the min-ttl and max-ttl parameters are configured accordingly. However, the T-PE/S-PE node still probes all hops up to min-ttl to correctly build the FEC of the needed subset of segments.

Note that this method does not require the use of the downstream mapping TLV in the echo request and echo reply messages.

### 3.1.31.1 VCCV for static pseudowire segments

MS pseudowire is supported with a mix of static and signaled pseudowire segments. However, VCCV ping and VCCV-trace is allowed until at least one segment of the MS pseudowire is static. Users cannot test a static segment but also, cannot test contiguous signaled segments of the MS-pseudowire. VCCV ping and VCCV trace is not supported in static-to-dynamic configurations.

### 3.1.31.2 Detailed VCCV-trace operation

[Figure 39: VCCV ping over a multi-segment pseudowire](#) shows how a trace can be performed on the MS-pseudowire originating from T-PE1 by a single operational command. The following process occurs:

1. T-PE1 sends a VCCV echo request with TTL set to 1 and a FEC 128 containing the pseudowire information of the first segment (pseudowire1 between T-PE1 and S-PE) to S-PE for validation.
2. S-PE validates the echo request with the FEC 128. Because it is a switching point between the first and second segment, it builds an echo reply with a return code of 8 and includes the FEC 128 of the second segment (pseudowire2 between S-PE and T-PE2) and sends the echo reply back to T-PE1.
3. T-PE1 builds a second VCCV echo request based on the FEC128 in the echo reply from the S-PE. It increments the TTL and sends the next echo request out to T-PE2. Note that the VCCV echo request packet is switched at the S-PE datapath and forwarded to the next downstream segment without any involvement from the control plane.
4. T-PE2 receives and validates the echo request with the FEC 128 of the pseudowire2 from T-PE1. Because T-PE2 is the destination node or the egress node of the MS-pseudowire it replies to T-PE1 with an echo reply with a return code of 3, (egress router) and no FEC 128 is included.
5. T-PE1 receives the echo reply from T-PE2. T-PE1 recognizes that T-PE2 is the destination of the MS pseudowire because the echo reply does not contain the FEC 128 and because its return code is 3. The trace process is completed.

### 3.1.31.3 Control plane processing of a VCCV echo message in a MS-pseudowire

#### 3.1.31.3.1 Sending a VCCV echo request

When in the ping mode of operation, the sender of the echo request message requires the FEC of the last segment to the target S-PE/T-PE node. This information can either be configured manually or be



obtained by inspecting the corresponding sub-TLV's of the pseudowire switching point TLV. However, the pseudowire switching point TLV is optional and there is no guarantee that all S-PE nodes populate it with their system address and the pseudowire ID of the last pseudowire segment traversed by the label mapping message. Thus, the router implementation always makes use of the user configuration for these parameters.

When in the trace mode operation, the T-PE automatically learns the target FEC by probing one by one the hops of the MS-pseudowire path. Each S-PE node includes the FEC to the downstream node in the echo reply message in a similar way that LSP trace causes the probed node to return the downstream interface and label stack in the echo reply message.

### 3.1.31.3.2 Receiving a VCCV echo request

Upon receiving a VCCV echo request the control plane on S-PEs (or the target node of each segment of the MS pseudowire) validates the request and responds to the request with an echo reply consisting of the FEC 128 of the next downstream segment and a return code of 8 (label switched at stack-depth) indicating that it is an S-PE and not the egress router for the MS-pseudowire.

If the node is the T-PE or the egress node of the MS-pseudowire, it responds to the echo request with an echo reply with a return code of 3 (egress router) and no FEC 128 is included.

### 3.1.31.3.3 Receiving a VCCV echo reply

The operation to be taken by the node that receives the echo reply in response to its echo request depends on its current mode of operation such as ping or trace.

In ping mode, the node may choose to ignore the target FEC 128 in the echo reply and report only the return code to the operator.

However, in trace mode, the node builds and sends the subsequent VCCV echo request with a incrementing TTL and the information (such as the downstream FEC 128) it received in the echo request to the next downstream pseudowire segment.

## 3.1.32 IGMP snooping diagnostics

### 3.1.33 MFIB ping

The multicast forwarding information base (MFIB) ping OAM tool allows to easily verify inside a VPLS which SAPs would normally egress a specific multicast stream. The multicast stream is identified by a source unicast and destination multicast IP address, which are mandatory when issuing an MFIB ping command.

An MFIB ping packet is sent through the data plane and goes out with the data plane format containing a configurable VC label TTL. This packet traverses each hop using forwarding plane information for next hop, VC label, and so on. The VC label is swapped at each service-aware hop, and the VC TTL is decremented. If the VC TTL is decremented to 0, the packet is passed up to the management plane for processing. If the packet reaches an egress node, and would be forwarded out a customer facing port (SAP), it is identified by the OAM label below the VC label and passed to the management plane.

### 3.1.34 MPLS-TP on-demand OAM commands

Ping and trace tools for PWs and LSPs are supported with both IP encapsulation and the MPLS-TP on demand CV channel for non-IP encapsulation (0x025).

### 3.1.35 MPLS-TP pseudowires: VCCV ping/VCCV trace

The 7450 ESS, 7750 SR, and 7950 XRS routers support VCCV Ping and VCCV Trace on single segment PWs and multi-segment PWs where every segment has static labels and a configured MPLS-TP PW Path ID. It also supports VCCV Ping and Trace on MS-PWs here a static MPLS-TP PW segment is switched to a dynamic T-LDP signaled segment.

Static MS-PW PWs are referred to with the sub-type static in the vccv-ping and vccv-trace command. This indicates to the system that the rest of the command contains parameters that are applied to a static PW with a static PW FEC.

Two ACH channel types are supported: the IPv4 ACH channel type, and the non-IP ACH channel type (0x0025). This is known as the non-ip associated channel. This is the default for type static. The Generic ACH Label (GAL) is not supported for PWs.

If the IPv4 associated channel is specified, then the IPv4 channel type is used (0x0021). In this case, a destination IP address in the 127/8 range is used, while the source address in the UDP/IP packet is set to the system IP address, or may be explicitly configured by the user with the **src-ip-address** option. This option is only valid if the IPv4 control-channel is specified.

The reply mode is always assumed to be the same application level control channel type for type static.

As with other PW types, the downstream mapping and detailed downstream mapping TLVs (DSMAP/DDMAP TLVs) are not supported on static MPLS-TP PWs.

The follow CLI command description shows the options that are only allowed if the type static option is configured. All other options are blocked.

```
vccv-ping static sdp-id:vc-id [target-fec-type pw-id-fec sender-src-address ip-addr remote-dst-address ip-address pw-id pw-id pw-type pw-type] [dest-global-id global-id dest-node-id node-id] [assoc-channel ipv4 | non-ip] [fc fc-name [profile {in | out}]] [size octets] [count send-count] [timeout timeout] [interval interval] [ttl vc-label-ttl] [src-ip-address ip-addr]
```

```
vccv-trace static sdp-id:vc-id [assoc-channel ipv4 | non-ip] [src-ip-address ipv4-address] [target-fec-type pw-id sender-src-address ip-address remote-dst-address ip-address pw-id pw-id pw-type pw-type] [detail] [fc fc-name [profile in | out]] [interval interval-value] [max-fail no-response-count] [max-ttl max-vc-label-ttl] [min-ttl min-vc-label-ttl] [probe-count probe-count] [size octets] [timeout timeout-value]
```

If the spoke SDP referred to by the *sdp-id:vc-id* has an MPLS-TP PW-Path-ID defined, then those parameters are used to populate the static PW TLV in the target FEC stack of the **vccv-ping** or **vccv-trace** packet. If a Global-ID and Node-ID is specified in the command, then these values are used to populate the destination node TLV in the **vccv-ping** or **vccv-trace** packet.

The global-id/node-id are only used as the target node identifiers if the vccv-ping is not end-to-end (for example, a TTL is specified in the **vccv-ping** or **trace** command and it is < 255), otherwise the value in the PW Path ID is used. For **vccv-ping**, the *dest-node-id* may be entered as a 4-octet IP address <a.b.c.d> or 32-bit integer <1 to 4294967295>. For **vccv-trace**, the destination node-id and global-id are taken from the spoke SDP context.

The same command syntax is applicable for SAA tests configured under `configure saa test a type`.

### 3.1.35.1 VCCV ping and VCCV trace between static MPLS-TP and dynamic PW segments

The 7450 ESS, 7750 SR, and 7950 XRS routers support end to end VCCV Ping and VCCV trace between a segment with a static MPLS-TP PW and a dynamic T-LDP segment by allowing the user to specify a target FEC type for the VCCV echo request message that is different from the local segment FEC type. That is, it is possible to send a VCCV Ping / Trace echo request containing a static PW FEC in the target stack TLV at a T-PE where the local egress PW segment is signaled, or a VCCV Ping or Trace echo request containing a PW ID FEC (FEC128) in the target stack TLV at a T-PE where the egress PW segment is a static MPLS-TP PW.

Note that all signaled T-LDP segments and the static MPLS-TP segments along the path of the MS-PW must use a common associated channel type. Because only the IPv4 associated channel is supported in common between the two segments, this must be used. If a user selects a non-IP associated channel on the static MPLS-TP spoke SDP, then **vccv-ping** and **vccv-trace** packets are dropped by the S-PE.

The **target-fec-type** option of the **vccv-ping** and **vccv-trace** command is used to indicate that the remote FEC type is different from the local FEC type. For a vccv-ping initiated from a T-PE with a static PW segment with MPLS-TP parameters, attempting to ping a downstream FEC128 segment, then a target-fec-type of pw-id is configured with a static PW type. In this case, an assoc-channel type of non-ip is blocked, and the other way around. Likewise the reply-mode must be set to control-channel. For a vccv-ping initiated from a T-PE with a FEC128 PW segment, attempting to ping a downstream static PW FEC segment, a target-fec-type of static is configured with a pw-id PW type, then a control-channel type of non-ip is blocked, and the other way around. Likewise the reply-mode must also be set to control-channel.

When using VCCV Trace, where the first node to be probed is not the first-hop S-PE. the initial TTL must be set to >1. In this case, the target-fec-type refers to the FEC at the first S-PE that is probed.

The same rules apply to the control-channel type and reply-mode as for the vccv-ping case.

### 3.1.36 MPLS-TP LSPs: LSP ping/LSP trace

For lsp-ping and lsp-trace commands:

- sub-type **static** must be specified. This indicates to the system that the rest of the command contains parameters specific to a LSP identified by a static LSP FEC.
- The 7450 ESS, 7750 SR, and 7950 XRS routers support the use of the G-ACh with non-IP encapsulation, IPv4 encapsulation, or labeled encapsulation with IP de-multiplexing for both the echo request and echo reply for LSP-Ping and LSP-Trace on LSPs with a static LSP FEC (such as MPLS-TP LSPs).
- It is possible to specify the target MPLS-TP MEP/MIP identifier information for LSP Ping. If the target global-id and node-id are not included in the lsp-ping command, then these parameters for the target MEP ID are taken from the context of the LSP. The **tunnel-number** <tunnel-num> and **lsp-num** <lsp-num> for the far-end MEP are always taken from the context of the path under test.

```
lsp-ping static <lsp-name>
[force]
[path-type [active|working|protect]]
[fc <fc-name> [profile {in | out}]]
[size <octets>]
[ttl <label-ttl>]
[send-count <send-count>]
[timeout <timeout>]
[interval <interval>]
[src-ip-address <ip-address>]
[dest-global-id <dest-global-id> dest-node-id dest-node-id]
```

```
[assoc-channel none | non-ip | ipv4][detail]
lsp-trace static <lsp-name>
[force]
[path-type [active|working|protect]
[fc <fc-name> [profile {in|out}]]
[max-fail <no-response-count>]
[probe-count <probes-per-hop>]
[size <octets>]
[min-ttl <min-label-ttl>]
[max-ttl <max-label-ttl>]
[timeout <timeout>]
[interval <interval>]
[src-ip-address <ip-address>]
[assoc-channel none | non-ip | ipv4]
[downstream-map-tlv <dsmap|ddmap>]
[detail]
```

The following commands are only valid if the sub-type **static** option is configured, implying that the lsp-name refers to an MPLS-TP tunnel LSP:

**path-type** - Values: active, working, protect. Default: active.

**dest-global-id** <global-id> **dest-node-id** <node-id> - Default: **to** global-id:node-id from the LSP ID.

**assoc-channel**: If this is set to none, then IP encapsulation over an LSP is used with a destination address in the 127/8 range. If this is set to ipv4, then IPv4 encapsulation in a G-ACh over an LSP is used with a destination address in the 127/8 range. The source address is set to the system IP address, unless the user specifies a source address using the **src-ip-address** option. If this is set to **non-ip**, then non-IP encapsulation over a G-ACh with channel type 0x00025 is used. This is the default for sub-type static. Note that the encapsulation used for the echo reply is the same as the encapsulation used for the echo request.

**downstream-map-tlv**: LSP Trace commands with this option can only be executed if the control-channel is set to none. The DSMAP/DDMAP TLV is only included in the echo request message if the egress interface is either a numbered IP interface, or an unnumbered IP interface. The TLV is not included if the egress interface is of type **unnumbered-mpls-tp**.

For **lsp-ping**, the **dest-node-id** may be entered as a 4-octet IP address in the format a.b.c.d, or as a 32-bit integer in the range of 1 to 4294967295. For **lsp-trace**, the destination node-id and global-id are taken from the spoke-sdp context.

The send mode and reply mode are always taken to be an application level control channel for MPLS-TP.

The **force** parameter causes an LSP ping echo request to be sent on an LSP that has been brought oper-down by Bi-directional Forwarding Detection (BFD) (LSP-Ping echo requests would normally be dropped on oper-down LSPs). This parameter is not applicable to SAA.

The LSP ID used in the LSP ping packet is derived from a context lookup based on lsp-name and path-type (active/working/protect).

**dest-global-id** and **dest-node-id** refer to the target global/node ID. They do not need to be entered for end-to-end ping and trace, and the system uses the destination global ID and node ID from the LSP ID.

The same command syntax is applicable for SAA tests configured under **config>saa>test**.

### 3.1.37 VXLAN ping supporting EVPN for VXLAN

EVPN is an IETF technology per RFC7432 that uses a new BGP address family and allows VPLS services to be operated as IP-VPNs, where the MAC addresses and the information to setup the flooding trees are distributed by BGP. The EVPN VXLAN connections, VXLAN Tunnel Endpoint (VTEP), uses a connection

specific OAM Protocol for on demand connectivity verification. This connection specific OAM tool, VXLAN Ping, is described in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 2 Services and EVPN Guide*, within the VXLAN Section.

### 3.1.38 Show commands

The outputs in the following section are examples only; actual displays may differ depending on supported functionality and user configuration.

### 3.1.39 BFD

The existing **show>router>bfd** is enhanced for MPLS-TP, as follows:

#### **show>router>bfd>mpls-tp-lsp**

This command displays the MPLS –TP paths for which BFD is enabled.

#### **show>router>bfd>session [src ip-address [dest ip-address | detail]] | [mpls-tp-path lsp-id... [detail]]**

This command shows the details of the BFD session on a particular MPLS-TP path, where lsp-id is the fully qualified lsp-id to which the BFD session is in associated.

An example output is shown below:

```
*A:mlstp-dutA# show router bfd
- bfd

bfd-template      - Display BFD Template information
interface         - Display Interfaces with BFD
session           - Display session information

*A:mlstp-dutA# show router bfd bfd-template "privatebed-bfd-template"
=====
BFD Template privatebed-bfd-template
=====
Template Name      : privatebed-* Template Type      : cpmNp
Transmit Timer    : 10 msec   Receive Timer      : 10 msec
CV Transmit Interval : 1000 msec
Template Multiplier : 3       Echo Receive Interval : 100 msec

Mpls-tp Association
privatebed-oam-template
=====
* indicates that the corresponding row element may have been truncated.
*A:mlstp-dutA# show router bfd session
=====
BFD Session
=====
Interface/Lsp Name      State      Tx Intvl  Rx Intvl  Multipl
Remote Address/Info    Protocols  Tx Pkts   Rx Pkts   Type
-----
wp::lsp-32              Down (1)   1000      1000      3
0::0.0.0.0             mplsTp    N/A       N/A       cpm-np
wp::lsp-33              Down (1)   1000      1000      3
0::0.0.0.0             mplsTp    N/A       N/A       cpm-np
wp::lsp-34              Down (1)   1000      1000      3
0::0.0.0.0             mplsTp    N/A       N/A       cpm-np
wp::lsp-35              Down (1)   1000      1000      3
0::0.0.0.0             mplsTp    N/A       N/A       cpm-np
```

```

wp::lsp-36          Down (1)          1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
wp::lsp-37          Down (1)          1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
wp::lsp-38          Down (1)          1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
wp::lsp-39          Down (1)          1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
wp::lsp-40          Down (1)          1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
wp::lsp-41          Down (1)          1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
pp::lsp-32          Up (3)           1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
pp::lsp-33          Up (3)           1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
pp::lsp-34          Up (3)           1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
pp::lsp-35          Up (3)           1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
pp::lsp-36          Up (3)           1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
pp::lsp-37          Up (3)           1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
pp::lsp-38          Up (3)           1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
pp::lsp-39          Up (3)           1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
pp::lsp-40          Up (3)           1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
pp::lsp-41          Up (3)           1000    1000    3
  0::0.0.0.0      mplsTp          N/A     N/A     cpm-np
-----
No. of BFD sessions: 20
-----
wp = Working path   pp = Protecting path
=====

```

### 3.1.40 MPLS Performance Monitoring (MPLS PM)

RFC 6374, *Packet Loss and Delay Measurement for MPLS Networks*, provides a standard packet format and process for measuring delay of a unidirectional or MPLS-TP using the General Associated Channel (G-ACh), channel type 0x000C. Unidirectional LSPs, such as RSVP-TE, require an additional TLV to return a response to the querier (the launch point). RFC 7876, *UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks*, defines the source IP information to include in the UDP Path Return TLV so the responding node can reach the querier using an IP network. The MPLS DM PDU does not natively include any IP header information. With MPLS TP there is no requirement for the TLV defined in RFC 7876.

The function of MPLS delay measurement is similar regardless of LSP type. The querier sends the MPLS DM query message toward the responder, transported in an MPLS LSP. The responder extracts the required PDU information to respond appropriately.

Launching MPLS DM tests are configured in the **config>oam-pm>session session-name test-family mpls** context. Basic architectural OAM PM components are required to be completed along with the MPLS specific configuration. The test PDU includes the following PDU settings:

- Channel Type: 0x000C (MPLS DM)

- Flags: Query
- Control Code: out-of-band (unidirectional LSP) and in-band (bidirectional LSP)
- Querier Timestamp Format: IEEE 1588-2008 (1588v2) Precision Time Protocol truncated timestamp format
- Session Identifier: The configured **oam-pm>session>mpls>dm test-id test-id**
- DSCP: The configured **oam-pm>session>mpls>dscp dscp-name** (this value is not used to convey or influence the CoS setting in the MPLS TC field, on the querier or reflector. The profile {**in | out**} and **fc fc-name** must be used to influence CoS markings at launch, and the MPLS TC influences CoS handling and marking upon reception.
- Timestamp 1: Set to the local transmit time in PTP format
- Timestamp 2 and 3: set to 0

TVLs can also be included, based on the configuration.

- **Padding (Type 0)**

Copy in Response: Padding is returned in the response when the **oam-pm>session>mpls>dm>reflect-pad** is configured.

- **Padding (Type 128)**

Do not copy in Response: Padding is not returned in the response when the **oam-pm>session>mpls>dm>pad-tlv-size** is configured without the **reflect-pad** command. This is the typical configuration with unidirectional LSPs.

- **UDP Return (Type 131)**

UDP Return object: The IP information used by the reflector to reach the querier for an out-of-band response, when the **oam-pm>session>mpls>lsp** is **rsvp** or **rsvp-auto** and the **udp-return-object** information is configured.

The maximum pad size of 257 is a result of the structure of the defined TLV. The length field is one byte, limiting the overall value filed to 255 bytes.

The reflector processes the inbound MPLS DM PDU and respond back to the querier based on the received information, using the response flag setting. Specific to the timestamp, the responder responds using the Query Timestamp Format, filling in the Timestamp 2 and Timestamp 3 values.

When the response arrives back at the querier, the delay metrics are computed. The common OAM-PM computation model and naming are used to simplify and rationalize the different technologies that leverage the OAM-PM infrastructure. The common methodology reports unidirectional and round trip delay metrics for Frame Delay (FD), InterFrame Delay Variation (IFDV), and Frame Delay Range (FDR). The term, "frame" is not indicative of the underlying technology being measured. It represents a normal cross technology naming with applicability to the appropriate naming for the measured technology. The common normal naming requires a mapping to the supported delay measurements included in RFC 6374.

*Table 9: Normalized naming mapping*

Description	RFC 6374	OAM-PM
A to B Delay	Forward	Forward
B to A Delay	Reverse	Backward

Description	RFC 6374	OAM-PM
Two Way Delay (regardless of processing delays within the remote endpoint B)	Channel	Round-Trip
Two Way Delay (includes processing delay at the remote endpoint B)	Round-Trip	Not supported

Because OAM-PM uses a common reporting model, unidirectional (forward and backward, and round trip) are always reported. With unidirectional LSPs, the T4 and T3 timestamps are zeroed but the backward and round trip directions are still reported. In the case of unidirectional LSPs, the backward and round trip values are of no significance for the measured MPLS network.

An MPLS DM test may measure the endpoints of the LSP when the TTL is set equal to or higher than the termination point distance. Midpoints along the path that support MPLS DM response functions can be queried by a test setting a TTL to expire along the path. The MPLS DM launch and reflection, including mid-path transit nodes, capability is disabled by default. To launch and reflect MPLS DM test packets **config>test-oam>mpls-dm** must be enabled.

The SR OS implementation supports MPLS DM Channel Type 0x000C from RFC 6374 function for the following:

- Label Switched Path types
  - RSVP-TE and RSVP-TE Auto LSPs set out-of-band response request and require the configuration of the UDP-Return-Object.
  - MPLS-TP sets in-band response request.
- Querier and Responder
- Traffic class indicator set to on (1)
- Querier Timestamp format PTP
- Mandatory TLV 0 – Copy padding
- Optional TLV 128 – Do not copy padding
- Optional TLV 131 – UDP-Return-Object (RFC 7876)

The following functions are not supported:

- Packet Loss measurement
- Throughput management
- Dyadic measurements
- Loopback
- Mandatory TLVs
  - TLV 1 – Return Address
  - TLV 2 – Session Query Interval
  - TLV 3 – Loopback Request
- Optional TLVs
  - TLV 129 – Destination Address



- TLV 130 – Source Address

### 3.1.40.1 Configuring MPLS PM

The following configuration provides an example that comprises the different MPLS OAM PM elements for the various LSPs. This example only includes configuration on the querier, excluding the basic MPLS and IP configurations. The equivalent MPLS configuration must be completed on all responders. Enabling MPLS DM is required on all queriers and responders.

#### 3.1.40.1.1 Accounting policy configuration

The following describes the accounting policy configuration:

```
config>log# info
-----
file-id 1
  description "OAM PM XML file Parameters"
  location cf2:
  rollover 10 retention 2
exit
accounting-policy 1
  description "Default OAM PM Collection Policy for 5-min Bins"
  record complete-pm
  collection-interval 5
  to file 1
  no shutdown
exit
log-id 1
exit
```

#### 3.1.40.1.2 Enabling MPLS DM

The following configuration enables EMPLD DM:

```
config>test-oam> #info
-----
mpls-dm
  no shutdown
exit
```

#### 3.1.40.1.3 RSVP LSP configuration

The following shows the RSVP LSP configuration:

```
config>router> #info
-----
mpls
  path "path-1"
  no shutdown
  exit
  lsp "LSP-PE-2-PE-1-via29" //lsp-name for oam-pm//
  to 192.0.2.1
```

```

    cspf
    include "via-P-3"
    primary "path-1"
    exit
  no shutdown
exit

```

### 3.1.40.1.4 RSVP-auto LSP configuration components

The following shows the RSVP-auto LSP configuration:

```

config>router> #info
-----
  mpls
    path "path-1"
      no shutdown
    exit

    lsp-template "auto-system-lsps" mesh-p2p //template name for oam-pm//
      from 1.1.1.31 //from address for oam-pm//
      default-path "path-1"
      cspf
      no shutdown
    exit

    auto-lsp lsp-template "auto-system-lsps" policy "auto-lsp"
      no shutdown
  exit
#-----
echo "Policy Configuration"
#-----
  policy-options
    begin
    prefix-list "mesh-p2p" //to addressing for oam-pm//
      prefix 1.1.1.28/32 exact
      prefix 1.1.1.29/32 exact
    exit
    policy-statement "auto-lsp"
      entry 10
        from
          prefix-list "mesh-p2p"
        exit
        action accept
      exit
    exit
  exit
  commit
exit

```

### 3.1.40.1.5 MPLS-TP LSP configuration

The following shows the MPLS-TP LSP configuration:

```

config>router> #info
-----
  mpls
    mpls-tp
      global-id 135

```

```

node-id 0.0.0.2
tp-tunnel-id-range 100 200
protection-template "ptcTemplate"
exit
oam-template "bfd-template"
bfd-template "bfdTemplate"
exit
no shutdown
exit
lsp "LSP-PE-2-PE-1-static" mpls-tp 100
to node-id 0.0.0.1
dest-global-id 135
dest-tunnel-number 100
working-tp-path
in-label 129
out-label 131 out-link "int-PE-2-P-3" next-hop 192.168.23.1
mep
oam-template "bfd-template"
bfd-enable cc
no shutdown
exit
no shutdown
exit
no shutdown
exit
no shutdown
exit

```

### 3.1.40.1.6 MPLS OAM-PM configuration

The following shows the MPLS OAM-PM configuration:

```

config>oam-pm# info
-----
bin-group 2 fd-bin-count 10 fdr-bin-count 10 ifdv-bin-count 10 create
bin-type fd
bin 1
lower-bound 1000
exit
bin 2
lower-bound 2000
exit
bin 3
lower-bound 3000
exit
bin 4
lower-bound 4000
exit
bin 5
lower-bound 5000
exit
bin 6
lower-bound 6000
exit
bin 7
lower-bound 7000
exit
bin 8
lower-bound 8000
exit
bin 9
lower-bound 9000
exit

```

```
exit
bin-type fdr
  bin 1
    lower-bound 1000
  exit
  bin 2
    lower-bound 1500
  exit
  bin 3
    lower-bound 2000
  exit
  bin 4
    lower-bound 2500
  exit
  bin 5
    lower-bound 3000
  exit
  bin 6
    lower-bound 3500
  exit
  bin 7
    lower-bound 4000
  exit
  bin 8
    lower-bound 4500
  exit
  bin 9
    lower-bound 5000
  exit
exit
bin-type ifdv
  bin 1
    lower-bound 500
  exit
  bin 2
    lower-bound 750
  exit
  bin 3
    lower-bound 1000
  exit
  bin 4
    lower-bound 1250
  exit
  bin 5
    lower-bound 1500
  exit
  bin 6
    lower-bound 1750
  exit
  bin 7
    lower-bound 2000
  exit
  bin 8
    lower-bound 2250
  exit
  bin 9
    lower-bound 2500
  exit
exit
no shutdown
exit

session "mpls-dm-rsvp-PE-2-PE-1" test-family mpls session-type proactive create
bin-group 2
```

```
description "mpls dm testing rsvp"
meas-interval 5-mins create
  accounting-policy 9
  event-mon
    delay-events
    no shutdown
  exit
exit
mpls
  dscp "af11"
  fc "af"
  lsp
    rsvp
      lsp "LSP-PE-2-PE-1-via29"
      udp-return-object 192.0.2.2
    exit
  exit
  profile in
  dm test-id 5 create
    interval 2000
    pad-tlv-size 257
    no shutdown
  exit
exit
session "mpls-dm-static-PE-2-PE-1" test-family mpls session-type proactive create
  bin-group 2
  description "mpls dm testing static-mpls-tp"
  meas-interval 5-mins create
  exit
  mpls
    dscp "af11"
    fc "af"
    lsp
      mpls-tp-static
      lsp "LSP-PE-2-PE-1-static"
    exit
  exit
  profile in
  ttl 5
  dm test-id 100 create
    interval 2000
    no shutdown
  exit
exit
session "mpls-dm-rsvp-auto-PE-2-PE-1" test-family mpls session-type proactive create
  bin-group 2
  description "mpls dm testing rsvp-auto-lsp"
  meas-interval 5-mins create
  exit
  mpls
    dscp "af11"
    fc "af"
    lsp
      rsvp-auto
      from 192.0.2.2
      lsp-template "auto-system-lsps"
      to 192.0.2.1
      udp-return-object 192.0.2.2
    exit
  exit
  profile in
  ttl 5
```

```
dm test-id 200 create
    interval 2000
    no shutdown
exit
exit
exit
```

### 3.1.41 BIER OAM

BIER supports the bier-ping and bier-trace OAM tools. FP4 and FP5 hardware is required and only IPv4 is supported. The tools are not supported:

- on VSR
- under the SAA tool or OAM-PM architecture
- MCI

A bier-ping packet is sent in a specific subdomain. The user can specify the subdomain in which the BIER OAM packet must be generated. In addition, the BIER OAM packet has to be destined for a BFER or a set of BFERs. Multiple BFERs can be specified for bier-ping. A single BFER can be specified for bier-trace. The BFER can be specified in one of the following ways:

- **through a BIER prefix**

The BIER prefix is flooded to the IGP domain through the IGP BIER TLV. The TLV also contains the BFR-ID, so the BIER prefix can be used to find the BFER BFR-ID and its corresponding SI and bit position in the BIER header. Up to 16 BIER prefixes can be specified for bier-ping.

- **through BFR-ID**

SR OS supports 16 SI and 255 bits for the BIER header, which means that 4K of BFR-ID are supported. A BFR-ID or a range of BFR-ID can be specified in the OAM command to build the SI and bit positions in the BIER header. Up to 16 contiguous BFR-IDs can be specified for bier-ping.

#### 3.1.41.1 ECMP and BIER OAM

ECMP is not supported for BIER in 7x50. For BIER OAM, Multipath Entropy data sub-TLV of Downstream Detailed Mapping TLV is used for ECMP discovery. SR OS does not support the Multipath Entropy data sub-TLV type. If SR OS receives Multipath Entropy data sub-TLV in the BIER OAM packet, it responds with the return code "One or more of the TLVs was not understood".

#### 3.1.41.2 Outbound time

BIER Ping and BIER Trace only support outbound time. Round-trip time is not supported, because multicast is unidirectional and BIER ping is in-band downstream, but out-of-band for echo reply. The outbound time is calculated from the network processor (NP) of the root to the NP of the leaf nodes, where the packet is timestamped.

#### 3.1.41.3 Negative outbound time

If negative outbound times display for BIER OAM, the cause is usually that the root and the leaf nodes are not synchronized. In this case, the user must ensure that the root and the leaf nodes are synchronized.

### 3.1.42 ICMP ping check connectivity checking using ICMP echo request and response

In specific network configurations, it is not always possible to deploy preferred standards-based purpose-built robust connectivity verification tools such as Bidirectional Forwarding Detection (BFD), or Ethernet Connectivity Fault Management Continuity Check Message (ETH-CCM), or other more suited tools. When circumstances prevent the preferred connectivity validation methods, ICMP echo request and response ping connectivity check (**icmp ping check**) using ping templates can be used as an alternate connectivity checking method. Before deploying this approach an understanding of the treatment of these types of packets on the involved network elements is required. The ping check affects the operational state of the VRPN or IES service IPv4 interface (the service IP interface) being verified.

Deployment of this feature requires the following:

- configuring the ping template
- the assignment of the ping template to a service IP interface
- optionally, configuring the distributed CPU protection for the **icmp-ping-check** protocol

The ping template defines timers and thresholds that determine the basis for connectivity verification and influence the service IP interface operational state. The configuration of the ping template is located in the **config>test-oam>icmp** context. The configuration options are separated to allow different failure detection and recovery behaviors.

The transmission frequency (**interval**), loss detection (**timeout**) and threshold (**failure-threshold**) are used to check connectivity when the service IP interface is steady and operationally up, or steady and operationally down. When these values are monitoring connectivity and the service IP interface is operationally up, consecutive failures that reach the failure threshold transitions the interface to operationally down. When these values are monitoring connectivity and the service IP interface is operationally down, a first success triggers the recovery values to complete the validation. For example, if the **interval 10** (seconds), **timeout 5** (seconds) and **failure-threshold 3** (count) the failure detection takes 30 seconds.

When a service IP interface transitions from operationally up to operationally down because of icmp ping check the log event "UTC WARNING: SNMP #2004 vprn1000999 int-PE-CE-999. Interface int-PE-CE-999 is not operational" is generated.

When a service IP interface has transitioned from operationally up to the operationally down state because of icmp ping check, the transmission continues at the specified interval until there is a successful ICMP echo response related to the ICMP echo request. When the first success is received, there is a possible transition from operationally down to operationally up and the function moves to the recovering phase. The icmp ping check packets for the affected service IP interface starts to transmit at frequency (**reactivation-interval**), invoking loss detection (**reactivation-timeout**) and consecutive success count (**reactivation-threshold**). If the reactivation threshold is reached, the service IP interface transitions from operationally down to operationally up. The transmission frequency (**interval**), loss detection (**timeout**) and threshold (**failure-threshold**) are used to monitor the service IP interface.

When a service IP interface transitions from operationally down to operationally up because of icmp ping check the log event "UTC WARNING: SNMP #2005 vprn1000999 int-PE-CE-999. Interface int-PE-CE-999 is operational" is generated.

If a failure occurs in the recovering phase, the **reactivation-failure-threshold** is consulted to determine the number of retries that should be attempted in this phase. This option allows a service IP interface a specified number of retries in this phase before returning to transmitting at **interval** and those associated values. The **reactivation-failure-threshold** parameter is bypassed if there was a previous success for the service IP interface in the recovery phase for the latest transition. This parameter determines the number

of consecutive failures, without a previous success, before declaring the recovering is not proceeding and returns to the interval values. In larger scale environments this value may need to be increased.

Only packets related to the icmp ping check, ICMP echo request and ARP packets specifically associated with the assigned local ping template, can be sent when the interface is operationally down because of an icmp ping check failure. Only packets related to the icmp ping check, ICMP echo response and ARP packets specifically associated with the assigned local ping template, can be received when the interface is operationally down because of the ping check failure.

A ping check function should never be configured on both peers. This leads to deadlock conditions that can only be resolved by manually disabling the ping template under the interface. As previously stated, only packets associated with the local ping template can be transmitted and received on a service IP interface when the interface is operationally down because of icmp check.

The configured ping template values can be updated without having to change the administrative state or existing references. However, the service IP interfaces that reference a specific ping template configuration imports the values when the ping-template is administratively enabled under the service IP interface. There is no automatic updating of modified ping template values on service IP interfaces referencing a ping-template. To push the changes to the referencing service IP interface the command **tools>perform>test-oam>icmp>ping-template-sync** *template-name* is available. This command updates all interfaces that reference the specified ping-template. Executing this command updates all the referencing service IP interfaces in the background after the command is accepted. If there is an HA event and the **tools** command has not completed updating, all the interfaces that had were not updated at the time of the HA event do not receive the new values. If an HA event occurs and there is a concern that all interfaces may not have received the update the command should be executed again on the newly active. The command does not survive an HA event.

For a service IP interface to import and start using the icmp ping check, the **ping-template** *template-name* must be enabled and the **destination-address** *ip-address*, must be configured. When the **ping-template** is added to the service IP interface the values associated with that **ping-template** are imported. When the ping-template's administrative state under the service IP interface is enabled, the values are checked again to ensure the latest values associated with the **ping-template** are being used. The source ip address of the packet is the primary IPv4 address of the service IP interface. This is not a configurable parameter.

When the **ping-template** command is administratively enabled under a service IP interface that is operationally up, the interface is assumed to have connectivity until proven otherwise. This means the interface state is not affected unless the ping template determines that there are connectivity issues based on the **interval**, **timeout**, and **failure-threshold** commands. If the wanted behavior is for the ping-template to validate service IP interface connectivity before allowing the service IP interface to become operational, the service IP interface can be administratively disabled, the ping-template enabled under that interface, and then the interface administratively enabled. This is considered to be operationally down because of underlying conditions.

When the **ping-template** command is administratively enabled under a service IP interface that is operationally down because of an underlying condition unrelated to icmp ping check, when the underlying condition is cleared, the icmp ping check prevents the interface from entering the operationally up state until it can verify the connectivity. When the underlying condition is cleared the icmp ping check function enters the recovering phase using the **reactivation-interval**, **reactivation-timeout**, **reactivation-threshold**, and the **reactivation-failure-threshold** values.

When a node is rebooted, service IP interfaces, with administratively enabled ping templates, must verify the interface connectivity before allowing it to progress to an operationally up state. This ensures that the interface does not bounce from operationally up to operationally down after a reboot and the service IP interface state is properly reflected when the reboot is complete. Service IP interfaces that have an administratively enabled **ping-template** enter the recovering phase using the **reactivation-interval**,



**reactivation-timeout**, **reactivation-threshold** and the **reactivation-failure-threshold** values following a reboot.

When a soft reset condition is raised icmp ping check state for the service IP interface is held in the same state it entered the process until the soft reset is complete. The interfaces exit the soft reset in the same phase they entered but all counters are cleared. The service IP interfaces that have an administratively enabled ping template enter this held state if they are in any way related to any hardware that is undergoing a soft reset. Two examples to demonstrate the expected behavior are shown below. When a service IP interface is related to a LAG, if a single port member in that LAG is affected by the soft reset, the interface enters this held state. Similarly, if the service IP interface is connected using an R-VPLS configuration it enters the held state.

The protocol used to determine the icmp ping check function has been added to the distributed CPU protection list of protocols, **icmp-ping-check**. The distributed CPU protection function can be used to limit the amount of icmp ping check packets received on a service IP interface with an enabled ping template. This is an optional configuration that would prevent crossover impact on unrelated service IP interfaces using icmp ping check because of a rogue interface.

The **show>service>id>interface ip-int-name detail** command has been updated with the **ping-template** values and operational information. The most effective way to view the output is to use a match criterion for "Ping Template Values in Use". The "Ping Template Values in Use" section of the output reports the current values that were imported from the referenced **config>test-oam>icmp>ping-template**. The "Operational Data" section of the output includes the administrative state (Up or Down) and destination address being tested (IP address or notConfigured). It also includes the current interval in use (interval or reactivation-interval) and the current state being reported, (operational, notRunning, failed). There are also pass and fail counters reporting, while in the current state, the number of consecutive passes or fails that have occurred. This provides a stability indicator. If these values are low, it may indicate that even though no operational state transitions have occurred there are intermittent but frequent failures. If neither of these counter are incrementing it is likely an underlying condition has been detected and the icmp ping check is not attempting to send and cannot receive connectivity packets. These counters are cleared when moving between different intervals, and for a soft reset.

```
show service id <service-id> interface <ip-int-name> detail | match "Ping Template Values in
Use" post-lines 29
Ping Template Values in Use
Name                : customer-access-basic
Description         : basic service detection and recovery
Dscp                : ncl
Dot1p               : 7
Interval            : 10
Timeout             : 1
Failure Threshold: 3
React Fail Thresh: 9
React Interval     : 1
React Timeout      : 1
React Threshold    : 3
Size                : 56
TTL                 : 1
Ping Template Operational Data
Admin State        : Up
Destination       : 192.9.99.2
Current Interval  : Interval
Current State     : Operational
Ping Template Counters
Fail Counter      : 0
Pass Counter      : 107
```

The **show>test-oam>icmp>ping-template** and **show>test-oam>icmp>ping-template-using** have been added to display the various **config>test-oam>icmp>ping-template** configurations and services referencing the ping templates.

Using icmp ping check enabled on service IP interfaces incur longer recovery delays on failure and reboot because of the additional validations required to validate those interfaces.

The icmp ping check function supports IPv4 interfaces created on SAPs in VRPN and IES services and R-VPLS services, as well as Ethernet satellite (esat) connections. When the service IP interface is making use of an R-VPLS configuration, the interface between the VRPN or IES service and the VPLS service is a virtual connection. In order for the icmp ping check to function properly in R-VPLS environments, the connection being used to validate the peer must be reachable over a SAP.

The icmp ping check should only be used when other purpose-built connectivity checking is not a deployable solution. Interaction with contending protocols may be unexpended.

The interaction between icmp ping check and service IP interface hold-time, in general, the **hold-time up** option delays the deactivation of the associated IP interface by the specified number of seconds. The **hold-time down** option delays the activation of the associated IP interface by the specified number of seconds.

With the **hold-time up** option, if a service IP interface is about to transition from operational up to down because the port transitioned from operational up to down, loss of signal, administrative down, and so on, then **hold-time up** timer is started. The interface remains operationally up until the timer expires. The icmp ping check runs in parallel because the underlying operational state has been delayed. If it lasts longer than the detection for the icmp ping check it could fail while the interval is counting down. If the **hold-time up** counter expires the interface transitions to operationally down and the icmp ping check now recognizes the underlying issue and stops trying to transmit. Normal underlying condition recovery noted earlier in this section follow.

If however, the **hold-time up** is short circuited because the port returns to an operationally up state before the expiration of the **hold-time up**, the following interactions are noted:

- If the icmp ping check has not failed before the port returns to operational up, the service IP interface stays operational and the icmp ping check continues at interval without ever have affecting the operational state of the interface.
- If the icmp ping check has registered a failure during this time the service IP interface transitions to operational down because of the icmp ping check and the icmp ping check must recover the interface using the reactivation-interval.

With the **hold-time down** option, if a service IP interface is about to transition from operationally down to up because the port transitioned from operationally down to up, the interface remains down until the expiration of the down timer. When the timer expires, the icmp ping check follows the normal underlying condition recovery noted earlier in this section follows.

These validations do not support or impact IPv6 interfaces.

There is no support for **config>system>enable-icmp-vse** Nokia-specific ICMP packets on interfaces that are using ping templates.

ICMP ping check connectivity is only supported on FP3-based and above platforms and should not be configured on any service IP interfaces that are configured over hardware that does not meet this requirement.

## 3.2 IP Performance Monitoring (IP PM)

SR OS supports Two-Way Active Measurement Protocol (TWAMP) and Two-Way Active Measurement Protocol Light (TWAMP Light) and Simple Two-Way Active Measurement Protocol (STAMP).

### 3.2.1 TWAMP

TWAMP provides a standards-based method for measuring the IP performance (packet loss, delay, and jitter) between two devices. TWAMP leverages the methodology and architecture of One-Way Active Measurement Protocol (OWAMP) to define a way to measure two-way or round-trip metrics.

There are four logical entities in TWAMP: the Control-Client, the Session-Sender, the server, and the Session-Reflector. The Control-Client and Session-Sender are typically implemented in one physical device (the "client") and the server and Session-Reflector in a second physical device (the "server"). The router acts as the "server".

The Control-Client and server establish a TCP connection and exchange TWAMP-Control messages over this connection. When a server accepts the TCP control session from the Control-Client, it responds with a server greeting message. This greeting includes the various modes supported by the server. The modes are in the form of a bit mask. Each bit in the mask represents a functionality supported on the server. When the Control-Client wants to start testing, the client communicates the test parameters to the server, requesting any of the modes that the server supports. If the server agrees to conduct the described tests, the test begins as soon as the Control-Client sends a Start-Sessions or Start-N-Session message. As part of a test, the Session-Sender sends a stream of UDP-based TWAMP test packets to the Session-Reflector, and the Session-Reflector responds to each received packet with a UDP-response TWAMP test packet. When the Session-Sender receives the response packets from the Session-Reflector, the information is used to calculate two-way delay, packet loss, and packet delay variation between the two devices. The exchange of TWAMP test PDUs is referred to as a TWAMP-Test.

The TWAMP test PDU does not achieve symmetrical packet size in both directions unless the frame is padded with a minimum of 27 bytes. The Session-Sender is responsible for applying the required padding. After the frame is appropriately padded, the Session-Reflector reduces the padding by the number of bytes needed to provide symmetry.

Server mode support includes:

- individual session control (Mode Bit 4: Value 16)
- reflected octets (Mode Bit 5: Value 32)
- symmetrical size test packet (Mode Bit 6: Value 64)

### 3.2.2 TWAMP Light and STAMP

#### 3.2.2.1 Overview



**Note:** For consistency within the SR OS, the **twamp-light** command sends the IP PM packet without a direct correlation to the actual test PDU format, TWAMP Light, or STAMP. Similarly, the documentation aligns with that nomenclature.

TWAMP Light was introduced as part of RFC 5357, *A Two-Way Active Measurement Protocol (TWAMP), Appendix I (Informational)*. The RFC appendix defined a single-ended test without the requirement to use

the TCP control channel over which the Control-Client and server negotiate test parameters. Using this approach, configuration on both entities, the Session-Sender and the Session-Reflector, replaces the control channel and provides the application-specific handling information required to launch and reflect test packets. In other words, TWAMP Light uses the TWAMP test packet for gathering IP performance information, but eliminates the need for the TWAMP TCP control channel. However, not all negotiated control parameters are replaced with local configuration. For example, QoS parameters communicated over the TWAMP control channel are replaced with a reply-in-kind approach for TWAMP Light. The reply-in-kind model reflects back the received QoS parameters, which can be influenced by the QoS policies of the Session-Reflector.

This informational work formed the baseline for standardization of TWAMP Light by RFC 8762, *Simple Two-Way Active Measurement Protocol (STAMP)*. The STAMP standard defined by RFC 8762 is backward compatible, as described in the RFC 5357 appendix. As the STAMP work in the IETF continues to evolve, the backward compatibility has remained largely unchanged. However, RFC 8972, *Simple Two-Way Active Measurement Protocol Optional Extension*, introduces advanced capabilities to the base version of the protocol. This creates areas to consider.

The handling of some functions that were open to interpretation in the original TWAMP Light appendix is formalized in RFC 8972. To properly handle these mismatches, the SR OS uses a function that allows the user to define the wanted behavior for the Session-Sender (under the **configure oam-pm** context) and the Session-Reflector. Validation of configuration options are performed, and inconsistent configuration based on protocol handling is blocked.

Use the following command to configure the test packet type for the Session-Sender.

```
configure oam-pm session ip twamp-light session-sender-type
```

Use the following command to configure the test packet processing behavior for the Session-Reflector.

```
configure router twamp-light reflector type
```

The Session-Sender under the OAM-PM context uses the **twamp-light** and **stamp** options to determine the packet structure to transmit. For backward compatibility, the default packet format transmitted is **twamp-light**. When the **twamp-light** option is selected, any TLVs specific to STAMP test packets cannot be configured, including the **pad-tlv-size**. A Session-Sender should use the default pattern of 0 when communicating with a Session-Reflector acting in STAMP mode, to avoid conflicts when identifying the type of test packet arriving on the reflector. When the **stamp** option is selected, configuration options specific to TWAMP Light test packets cannot be configured, including the **pad-size** and a non-zero **pattern**.

The Session-Reflector uses the **twamp-light** and **stamp** options to determine its processing behavior for the test packet received from the Session-Sender. For backward compatibility, the default processing behavior is **twamp-light**. This type performs no TLV processing, treating any non-base packet octets as padding. If the required behavior for the Session-Reflector is to parse and process STAMP TLVs, the **stamp** command option must be used. Using this configuration, the Session-Reflector can accommodate both TWAMP Light and STAMP Session-Senders, processing the packet based on the TLV rules as defined in RFC 8972. Any Session-Sender that is transmitting TWAMP-Light-formatted test packets with additional padding must use an all-zero pattern to avoid ambiguity on the Session-Reflector.

The following describes PDU padding and the identification of STAMP TLVs.

- **TWAMP Light**

The TWAMP Light test packet request and response sizes are asymmetrical by default. The Session-Sender packet and the Session-Reflector packets are different sizes. To allow for symmetrical packets on the wire and packet size manipulation, the Session-Sender can configure the **pad-size octets**

command to increase the size of the packet. These octets are added directly to the base packet. The default pattern of the padding is all zeros which can be changed using the **pattern** command.

- **STAMP**

The STAMP packet request and response sizes are symmetrical by default. RFC 8762 defines a structured packet that ensures this behavior. To allow for general packet size manipulation, the STAMP Optional Extensions RFC 8972 defines a PAD TLV. This TLV is added after the base packet. STAMP padding uses the **pad-tlv-size octets** command to increase the size of the packet. An all-zero PAD pattern must be used for the PAD TLV.

The Session-Reflector uses configuration option **type twamp-light | stamp** to determine its processing behavior for the test packet received from the Session-Sender. For backward compatibility, the default processing behavior is **twamp-light**. This type performs no TLV processing, treating any non-base packet octets as padding. If the required behavior for the Session-Reflector is to parse and process STAMP TLVs, **type stamp** must be used. Using this configuration, the Session-Reflector can accommodate both TWAMP Light and STAMP Session-Senders, processing the packet based on the TLV rules defined by the STAMP Optional Extensions RFC 8972. Any Session-Sender that is transmitting TWAMP Light-formatted test packets with additional padding must use an all-zero pattern to avoid ambiguity on the Session-Reflector.

SR OS was an early adopter of IP Performance Measurement (IP PM) under the OAM Performance Monitoring (OAM-PM) architecture implementing TWAMP Light RFC 5357 Appendix I behavior. However, subsequent features introduced to the SR OS have adopted the standardized version STAMP and STAMP Optional Extensions.

In Link Measurement, the Session-Sender uses STAMP formatted packets.

In OAM-PM, the Session-Sender allows for a choice of TWAMP Light- or STAMP-formatted packets.

### 3.2.2.2 TWAMP Light Session-Reflector

The Session-Reflector receives and processes TWAMP Light test packets.

Use the following context to configure the Session-Reflector functions for base router reflection.

```
configure router twamp-light
```

Use the following context to configure Session-Reflector functions for per VPRN reflection.

```
configure service vprn twamp-light
```

The TWAMP Light Session-Reflector function is configured per context and must be activated before reflection can occur; the function is not enabled by default for any context. The Session-Reflector requires the user to define the TWAMP Light UDP listening port that identifies the TWAMP Light protocol. All the prefixes that the reflector accepts as valid sources for a TWAMP Light request must also be configured. If the source IP address in the TWAMP Light test packet arriving on the server does not match the configured prefixes, the packet is dropped. Multiple prefix entries may be configured per context on the server. Configured prefixes can be modified without shutting down the reflector function.



**Note:**

The TWAMP Light Session-Reflector **udp-port** *udp-port-number* range configured as part of the **config>service>twamp-light** and **router>twamp-light create** commands implement a restricted, reserved UDP port range that must adhere to a range of 862, 64364 to 64373 before an upgrade or reboot operation. Configurations outside this range result in a failure of the TWAMP Light Session-Reflector or prevent the upgrade operation. If an In-Service Software

Upgrade (ISSU) function is invoked when the **udp-port** *udp-port-number* is outside the allowable range and the TWAMP Light Session-Reflector is in a **no shutdown** state, the ISSU operation cannot proceed. The user must, at a minimum, disable the TWAMP Light Session-Reflector to allow the ISSU to proceed; however, the TWAMP Light Session-Reflector is not allowed to be enabled until the allowable range is met. A non-ISSU upgrade can proceed regardless of the state (enabled or disabled) of the TWAMP Light Session-Reflector. The configuration can load, however, the TWAMP Light Session-Reflector remains inactive following the reload when the allowable range is not met. When the **udp-port** *udp-port-number* for a TWAMP Light Session-Reflector is modified, all tests using the services of that reflector must update the **dest-udp-port** *udp-port-number* configuration parameter to match the new reflector listening port.

The TWAMP-Light Session-Reflector is stateful and supports unidirectional synthetic loss detection. An inactivity timeout under the **config>oam-test>twamp>twamp-light** command hierarchy defines the amount of time the TWAMP-Light Session-Reflector maintains individual test session in the absence of the arrival of test packets.

The TWAMP-Light Session-Reflector responds using the timestamp format that is indicated in the test packet from the Session-Sender. The Error Estimate Field is a two-byte field that includes an optional Z bit to indicate the format of timestamp. The TWAMP-Light Session-Reflector checks this field and replies using the same format for timestamp two (T2) and timestamp three (T3). The TWAMP-Light Session-Reflector does not interrogate or change the other bits in the Error Estimate field. Except for the processing of the Z bit, the received Error Estimate is reflected back to the Session-Sender.

Configurations that require an IPv6 UDP checksum of zero are increasing. In some cases, hardware timestamping functions that occur in the UDP header occur after the computation of the UDP checksum. Typically, packets that arrive with an IPv6 UDP checksum of zero are discarded. However, an optional configuration command **allow-ipv6-udp-checksum-zero** allows those packets to be accepted and processed for the configured UDP port of the TWAMP Light Session-Reflector.

Multiple tests sessions between peers are allowed. These test sessions are unique entities and may have different properties. Each test generates TWAMP test packets specific to their configuration. The TWAMP Light Session-Reflector includes the SSID defined by RFC 8972 as a fifth element augmenting the source IP, destination IP, source UDP port, and destination UDP port when maintaining the test state.

As TWAMP Light evolved, the TWAMP Light Session-Reflector required a method to determine processing of the arriving Session-Sender packets. The default processing behavior is type **twamp-light**. This treats all additional bytes beyond the base TWAMP Light packet as padding. The type **stamp** attempts to locate STAMP TLVs defined by RFC 8972 for processing.

See [Link measurement](#) and [OAM performance monitoring](#) for more information about the integration of TWAMP Light in those applications.

## 3.3 MPLS PM

### 3.3.1 TWAMP Light delay and loss for MPLS tunnels

The SR OS supports using TWAMP Light to measure the base router MPLS tunnel types in the following context.

```
configure oam-pm session ip tunnel mpls
```

See [TWAMP Light and STAMP](#) for more information about TWAMP Light.

When a TWAMP Light test configuration points to an MPLS tunnel, the complete test PDU is encapsulated in the MPLS tunnel and carried to the termination point of the tunnel based on MPLS forwarding rules. The **session** command must be configured to use the **ip** test family to allow for this mapping. The test family describes the underlying protocol used for testing, not the transport.

The following basic TWAMP Light IP configuration rules apply to the Session-Sender:

- The source IP address must be part of the base router route table on the Session-Sender.
- The destination must be an IP address in the base router on the terminating node of the MPLS tunnel where the Session-Reflector is configured.
- The destination UDP port must be the listening UDP port of the Session-Reflector terminating the MPLS tunnel.
- A Session-Reflector is required on the terminating node of the MPLS tunnel configured in the base router.

Use the commands in the following context to direct test packets to an MPLS transport.

```
configure oam-pm session ip tunnel
```

Users can specify the MPLS tunnel type to be used and the specific tunnel to carry the test packets. Entering a specific MPLS tunnel type sets it as the active configuration and deletes any configurations of different types under this context.

In the forward direction, test packets are encapsulated in the MPLS tunnel that matches the configuration. Because MPLS paths are unidirectional, the Session-Reflector performs an IP lookup in the base router route table and returns the packet using IP routing. This means IP reachability is required between the Session-Reflector and Session-Sender. Because the measurement is unidirectional delay, clock synchronization is required using PTP or an equivalently accurate time-distribution method. NTP does not have the accuracy to reliably produce unidirectional measurements.

IP MPLS tunnel configurations and IP forwarding configurations are mutually exclusive under the following context:

```
configure oam-pm session ip forwarding
```

Either type of test PDU (**twamp-light** or **stamp**) can be used for MPLS tunnel PM testing. The OAM-PM infrastructure, binning, delay streaming, threshold alarms, delay, and loss statistics are available for this testing.

Test statistics on the Session-Sender should be disregarded when an MPLS tunnel carrying the test packets recomputes on the Session-Sender node. Statistics during the MPLS tunnel change are not representative of the time to converge. Test convergence adds additional seconds on top of the tunnel recovery. This testing methodology is used to measure steady-state MPLS tunnel performance, not convergence, at the head of the tunnel.

### 3.3.2 RFC 6374 delay for MPLS tunnels

RFC 6374, *Packet Loss and Delay Measurement for MPLS Networks*, provides a standard packet format and process for measuring delay of a unidirectional or bidirectional label switched path (LSP) using the General Associated Channel (G-ACh), channel type 0x000C. Unidirectional LSPs, such as RSVP-TE, require an additional TLV to return a response to the querier (the launch point). RFC 7876, *UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks*, defines the source IP information to include in the UDP Path Return TLV so the responding node can reach the querier using an IP network.

The MPLS DM PDU does not natively include any IP source information. With MPLS TP there is no requirement for the TLV defined in RFC 7876.

The function of MPLS delay measurement is similar regardless of LSP type. The querier sends the MPLS DM query message toward the responder, transported in an MPLS LSP. The responder extracts the required PDU information to respond appropriately.

Launching MPLS DM tests is configured in the **config>oam-pm>session session-name test-family mpls** context. Basic architectural OAM PM components are required to be completed along with the MPLS specific configuration. The test PDU includes the following PDU settings;

For the base PDU:

- Channel Type: 0x000C (MPLS DM)
- Flags: Query
- Control Code: out-of-band (unidirectional LSP) and in-band (bidirectional LSP)
- Querier Timestamp Format: IEEE 1588-2008 (1588v2) Precision Time Protocol truncated timestamp format
- Session Identifier: The configured **oam-pm>session>mpls>dm test-id test-id**
- DSCP: The configured **oam-pm>session>mpls>dscp dscp-name** (this value is not used to convey or influence the CoS setting in the MPLS TC field. The profile {**in | out**} and **fc fc-name** must be used to influence CoS markings.
- Timestamp 1: Set to the local transmit time in PTP format
- Timestamp 2 and 3: set to 0

TVLs can also be included, based on the configuration.

- **Padding (Type 0)**

Copy in Response: Padding is returned in the response when the **oam-pm>session>mpls>dm>reflect-pad** is configured.

- **Padding (Type 128)**

Do not copy in Response: Padding is not returned in the response (the typical configuration with unidirectional LSPs) when the **oam-pm>session>mpls>dm>pad-tlv-size** is configured without the **reflect-pad** command.

- **UDP Return (Type 131)**

UDP Return object: The IP information used by the reflector to reach the querier for an out-of-band response, when the **oam-pm>session>mpls>lsp** is **rsvp** or **rsvp-auto** and the **udp-return-object** information is configured.

The maximum pad size of 257 is a result of the structure of the defined TLV. The length field is one byte, limiting the overall value to 255 bytes.

The reflector processes the inbound MPLS DM PDU and respond back to the querier based on the received information, using the response flag setting. Specific to the timestamp, the responder responds to the Query Timestamp Format, filling in the Timestamp 2 and Timestamp 3 values.

When the response arrives back at the querier, the delay metrics are computed. The common OAM-PM computation model and naming is used to simplify and rationalize the different technologies that leverage the OAM-PM infrastructure. The common methodology reports unidirectional and round trip delay metrics for Frame Delay (FD), InterFrame Delay Variation (IFDV), and Frame Delay Range (FDR). The term, "frame" is not indicative of the underlying technology being measured. It represents a normal cross



technology naming with applicability to the appropriate naming for the measured technology. The common normal naming requires a mapping to the supported delay measurements included in RFC 6374.

Table 10: Normalized naming mapping

Description	RFC 6374	OAM-PM
A to B Delay	Forward	Forward
B to A Delay	Reverse	Backward
Two Way Delay (regardless of processing delays within the remote endpoint B)	Channel	Round-Trip
Two Way Delay (includes processing delay at the remote endpoint B)	Round-Trip	—

Because OAM-PM uses a common reporting model, unidirectional (forward and backward), round-trip is always reported. With unidirectional measurements, the T4 and T3 timestamps are zeroed but the round-trip and backward direction are still reported. With unidirectional measurements, the backward and round trip values are not of any significance.

An MPLS DM test may measure the endpoints of the LSP when the TTL is set to or higher than the termination point. Midpoints along the path that support MPLS DM response functions can be targeted by a test by setting a TTL to expire along the path. The MPLS DM launch and reflection, including mid-path transit nodes, capability is disabled by default. To launch and reflect MPLS DM test packets **config>test-oam>mpls-dm** must be enabled.

The SR OS implementation supports the following MPLS DM Channel Type 0x000C from RFC 6374 function:

- Label Switched Path types
  - RSVP-TE and RSVP-TE Auto LSPs: sets out-of-band response request and requires the configuration of the UDP-Return-Object
  - MPLS-TP: sets in-band response request
- Querier and Responder
- Traffic class indicator set to on (1)
- Querier Timestamp format PTP
- Mandatory TLV 0 – Copy padding
- Optional TLV 128 – Do not copy padding
- Optional TLV 131 – UDP-Return-Object (RFC 7876)

The following functions are not supported:

- Packet Loss measurement
- Throughput management
- Dyadic measurements
- Loopback

- Mandatory TLVs
  - TLV 1 – Return Address
  - TLV 2 – Session Query Interval
  - TLV 3 – Loopback Request
- Optional TLVs
  - TLV 129 – Destination Address
  - TLV 130 – Source Address

### 3.4 ETH-CFM

The IEEE and the ITU-T have cooperated to define the protocols, procedures and managed objects to support service based fault management. Both IEEE 802.1ag standard and the ITU-T Y.1731 recommendation support a common set of tools that allow operators to deploy the necessary administrative constructs, management entities and functionality, Ethernet Connectivity Fault Management (ETH-CFM). The ITU-T has also implemented a set of advanced ETH-CFM and performance management functions and features that build on the proactive and on demand troubleshooting tools.

CFM uses Ethernet frames and is distinguishable by ether-type 0x8902. In specific cases, the different functions use a reserved multicast Layer 2 MAC address that could also be used to identify specific functions at the MAC layer. The multicast MAC addressing is not used for every function or in every case. The Operational Code (OpCode) in the common CFM header is used to identify the PDU type carried in the CFM packet. CFM frames are only processed by IEEE MAC bridges.

IEEE 802.1ag and ITU-T Y.1731 functions that are implemented are available on the SR and ESS platforms.

This section of the guide provides configuration example for each of the functions. It also provides the various OAM command line options and show commands to operate the network. The individual service guides provides the complete CLI configuration and description of the commands to build the necessary constructs and management points.

[Table 11: ETH-CFM acronym expansions](#) lists and expands the acronyms used in this section.

*Table 11: ETH-CFM acronym expansions*

Acronym	Expansion	Supported platform
1DM	One way Delay Measurement (Y.1731)	All
AIS	Alarm Indication Signal	All
BNM	Bandwidth Notification Message (Y.1731 sub OpCode of GNM)	All
CCM	Continuity check message	All
CFM	Connectivity fault management	All

Acronym	Expansion	Supported platform
CSF	Client Signal Fail (Receive)	All
DMM	Delay Measurement Message (Y.1731)	All
DMR	Delay Measurement Reply (Y.1731)	All
ED	Ethernet Defect (Y.1731 sub OpCode of MCC)	All
GNM	Generic Notification Message	All
LBM	Loopback message	All
LBR	Loopback reply	All
LMM	(Frame) Loss Measurement Message	Platform specific
LMR	(Frame) Loss Measurement Response	Platform specific
LTM	Linktrace message	All
LTR	Linktrace reply	All
MCC	Maintenance Communication Channel (Y.1731)	All
ME	Maintenance entity	All
MA	Maintenance association	All
MD	Maintenance domain	All
MEP	Maintenance association endpoint	All
MEP-ID	Maintenance association endpoint identifier	All
MHF	MIP half function	All
MIP	Maintenance domain intermediate point	All
OpCode	Operational Code	All
RDI	Remote Defect Indication	All
TST	Ethernet Test (Y.1731)	All
SLM	Synthetic Loss Message	All
SLR	Synthetic Loss Reply (Y.1731)	All
VSM	Vendor Specific Message (Y.1731)	All
VSR	Vendor Specific Reply (Y.1731)	All

### 3.4.1 ETH-CFM building blocks

The IEEE and the ITU-T use their own nomenclature when describing administrative contexts and functions. This introduces a level of complexity to configuration, discussion and different vendors naming conventions. The SR OS CLI has chosen to standardize on the IEEE 802.1ag naming where overlap exists. ITU-T naming is used when no equivalent is available in the IEEE standard. In the following definitions, both the IEEE name and ITU-T names are provided for completeness, using the format IEEE Name/ITU-T Name.

Maintenance Domain (MD)/Maintenance Entity (ME) is the administrative container that defines the scope, reach and boundary for testing and faults. It is typically the area of ownership and management responsibility. The IEEE allows for various formats to name the domain, allowing up to 45 characters, depending on the format selected. ITU-T supports only a format of "none" and does not accept the IEEE naming conventions.

- 0 is undefined and reserved by the IEEE.
- 1 indicates no domain name.
- 2,3, and 4 provide the ability to input various different textual formats, up to 45 characters. The string format (2) is the default and therefore the keyword is not shown when looking at the configuration.

Maintenance Association (MA)/Maintenance Entity Group (MEG) is the construct where the different management entities are contained. Each MA is uniquely identified by its MA-ID. The MA-ID comprises the MD level and MA name and associated format. This is another administrative context where the linkage is made between the domain and the service using the **bridging-identifier** configuration option. The IEEE and the ITU-T use their own specific formats. The MA short name formats (0 to 255) have been divided between the IEEE (0 to 31, 64 to 255) and the ITU-T (32 to 63), with five currently defined (1 to 4, 32). Even though the different standards bodies do not have specific support for the others formats a Y.1731 context can be configured using the IEEE format options.

The following formats are supported:

#### 1 (Primary VID)

values 0 to 4094

#### 2 (String)

raw ASCII, excluding 0-31 decimal/0-1F hex (which are control characters) from the ASCII table

#### 3 (2-octet integer)

values 0 to 65535

#### 4 (VPN ID)

Hex value as described in RFC 2685, *Virtual Private Networks Identifier*

#### 32 (icc-format)

exactly 13 characters from the ITU-T recommendation T.50



**Note:** When a VID is used as the short MA name, 802.1ag does not support VLAN translation because the MA-ID must match all the MEPs. The default format for a short MA name is an integer. Integer value 0 means the MA is not attached to a VID. This is useful for VPLS services on SR OS platforms because the VID is locally significant.



**Note:** The double quote character (") included as part of the ITU-T recommendation T.50 is not a supported character on the SR OS.

Maintenance Domain Level (MD Level)/Maintenance Entity Group Level (MEG Level) is the numerical value (0-7) representing the width of the domain. The wider the domain (higher the numerical value) the farther the ETH-CFM packets can travel. It is important to understand that the level establishes the processing boundary for the packets. Strict rules control the flow of ETH-CFM packets and are used to ensure correct handling, forwarding, processing and dropping of these packets. ETH-CFM packets with higher numerical level values flows through MEPs on MIPs on endpoints configured with lower level values. This allows the operator to implement different areas of responsibility and nest domains within each other. Maintenance association (MA) includes a set of MEPs, each configured with the same MA-ID and MD level used to verify the integrity of a single service instance.



**Note:** Domain format and requirements that match that format, as well as association format and those associated requirements, and the level must match on peer MEPs.

Maintenance Endpoints/MEG Endpoints (MEP) are the workhorses of ETH-CFM. A MEP is the unique identification within the association (1-8191). Each MEP is uniquely identified by the MA-ID, MEP-ID tuple. This management entity is responsible for initiating, processing and terminating ETH-CFM functions, following the nesting rules. MEPs form the boundaries which prevent the ETH-CFM packets from flowing beyond the specific scope of responsibility. A MEP has direction, **up** or **down**. Each indicates the directions packets are generated; **up** toward the switch fabric, **down** toward the SAP away from the fabric. Each MEP has an active and passive side. Packets that enter the active point of the MEP are compared to the existing level and processed accordingly. Packets that enter the passive side of the MEP are passed transparently through the MEP. Each MEP contained within the same maintenance association and with the same level (MA-ID) represents points within a single service. MEP creation on a SAP is allowed only for Ethernet ports with NULL, q-tags, q-in-q encapsulations. MEPs may also be created on SDP bindings. A vMEP is a service level MEP configuration that installs ingress (down MEP-like) extraction on the supported ETH-CFM termination points within a VPLS configuration.

Maintenance Intermediate Points/MEG Intermediate Points (MIPs) are management entities between the terminating MEPs along the service path. MIPs provide insight into the service path connecting the MEPs. MIPs only respond to Loopback Messages (LBM) and Linktrace Messages (LTM). All other CFM functions are transparent to these entities.

MIP creation is the result of the **mhf-creation** mode and interaction with related MEPs, and with the direction of the MEP. Two different authorities can be used to determine the MIPs that should be considered and instantiated. The domain and association or the **default-domain** hierarchies match the configured bridge identifier and VLAN to the service ID and any configured primary VLAN. When a primary VLAN MIP is not configured, the VLAN is either ignored or configured as **none**.

The domain and association MIP creation function triggers a search for all ETH-CFM domain association bridge identifier matches to the service it is linked to. A MIP candidate is then be evaluated using the **mhf-creation** mode and the rules that govern the algorithm. The domain association **mhf-creation** modes and their uses are listed below:

- **none**

A MIP is not a candidate for creation using this domain association bridge identifier. This is the default **mhf-creation** mode for every bridge identifier under this hierarchy.

- **explicit**

A MIP is a candidate for creation using this domain association bridge identifier only if a lower-level MEP exists.

- **default**

A MIP is a candidate for creation using this domain association bridge identifier regardless of the existence of a lower-level MEP. If a lower-level MEP is present, this creation mode behaves in the same manner as explicit creation mode.

- **static**

A MIP is a candidate for creation using the domain association bridge identifier at the level of the domain. This creation mode is specific to MIPs with the **primary-vlan-enabled** parameter configured. Different VLANs maintain their own level hierarchies. Primary VLAN creation under this context requires static mode.

For all modes except static mode, only a single MIP can be created. All candidates are collected and the lowest-level valid MIP is created. In static mode, all valid MIPs are created for the bridge identifier VLAN pair. A MIP is considered invalid if the level of the MIP is equal to or below a downward-facing MEP, or below the level of an upward-facing MEP and the MIP shares the same service component as the Up MEP.

Not all creation modes require the **mip** creation statement within the service. The explicit and default **mhf-creation** modes may instantiate a MIP without the **mip** creation statement under the service if a lower-level MEP exists for the domain association bridge identifier. If a lower-level MEP does not exist, the default and static **mhf-creation** modes require the **mip** creation statement on the service connection.

MEPs require the domain and association configurations to ensure that all ETH-CFM PDUs can be supported. MIPs have restricted ETH-CFM PDU support: ETH-LB and ETH-LT. These two protocols do not require the configuration of a domain and association. MIPs may be created outside of the association context using the default-domain table.

The **default-domain** table is an object table populated with values that are used for MIP creation. The table is indexed by the bridge identifier and VLAN. An index entry is automatically added when the **mip** creation statement is added under a SAP or SDP binding. When an index entry is added, the bridge identifier is set to the service ID and the VLAN is set to the **primary-vlan-enable** *vlan-id*. If the MIP does not use primary VLAN functionality, the VLAN is configured as **none**. When the entry has been added to the default-domain table, the default values can be configured. The default-domain table defers to the system-wide, read-only values.

Because there are two different locations able to process the MIP creation logic, a per-bridge identifier VLAN authority must be determined. The authority is a component, table, or configuration that is responsible for executing the MIP creation algorithm. In general, any domain association bridge identifier that could be used to create a specific MIP is authoritative. Other configurations influence the authority, such as the type of MIP (primary VLAN or non-primary VLAN), the different **mhf-creation** modes, the interaction of those modes with MEPs, and the direction of the MEP.

The following rules provide some high-level guidelines to determine the authority:

- **rule 1**

The original model predating the **default-domain** is always applied first. If a MIP is created using the original model, the new model is not applied. The original model includes complex Up MEP MIP creation rules. If an Up MEP exists on a service connection, any service connection other than the one with the active Up MEP attempts to create the lowest higher-level MIP using the domain association bridge identifier table. If a higher-level MIP cannot be created, and no higher-level association exists, the default-domain table is consulted.

- **rule 2**

A **mip** creation statement is required under the service connection to use the default-domain table. This is different from the domain association table. The domain association table does not require the **mip**

creation statement when the **mhf-creation** mode is configured as either explicit or default and a lower-level MEP is present.

- **rule 3**

If no domain association bridge identifier matches the service ID, the default-domain table is consulted.

- **rule 4**

If a domain association bridge identifier matches a service ID for the sole purpose of MEP creation, and no higher or lower domain association with the same bridge identifier exists, the default-domain table is consulted.

- **rule 4a**

Any domain association bridge identifier matching a service ID with a configured VLAN and a static **mhf-creation** mode is authoritative for all matching service IDs and MIPs with **primary-vlan-enable** configured with the same VLAN.

- **rule 4b**

Any domain association bridge identifier attempting to create a MIP with **primary-vlan-enable** configured is considered non-authoritative if the **mhf-creation** mode is anything other than static.

When the authority for MIP creation is determined, the MIP attributes are derived from that creation table. The default domain table defers to the read-only, system-wide MIP values and inherits those defaults. Some of the objects under the default-domain hierarchy must be configured using the same statement to avoid transient and unexpected MIP creation while the configuration is being completed. To this end, the **mhf-creation** mode and level have been combined in the same configuration statement.

The standard **mhf-creation** modes (**none**, **default**, **explicit**) are configurable as part of the default-domain table. Static mode can only be configured under the domain association bridge identifier. This is because default domain table indexing precludes multiple MIPs at different levels.

MIP creation requires configuration. The default values in both the domain association and the default domain table prevent MIP instantiation.

The **show eth-cfm mip-instantiation** command can be used to check the authority for each MIP.

There are two locations in the configuration where ETH-CFM is defined. The first location, where the domains, associations (including links to the service), MIP creation method, common ETH-CFM functions, and remote MEPs are defined under the top-level **eth-cfm** command. The second location is within the service or facility.

[Table 12: ETH-CFM support matrix](#) is a general table that indicates ETH-CFM support for the different services and SAP or SDP binding. It is not meant to indicate the services that are supported or the requirements for those services on the individual platforms.

*Table 12: ETH-CFM support matrix*

Service	Ethernet connection	Down MEP	Up MEP	MIP	Virtual MEP
Epipe	—	—	—	—	No
	SAP	Yes	Yes	Yes	—
	Spoke-SDP	Yes	Yes	Yes	—
	PW-SAP	No	No	Yes	—

Service	Ethernet connection	Down MEP	Up MEP	MIP	Virtual MEP
VPLS	—	—	—	—	Yes
	SAP	Yes	Yes	Yes	—
	Spoke-SDP	Yes	Yes	Yes	—
	Mesh-SDP	Yes	Yes	Yes	—
B-VPLS	—	—	—	—	Yes
	SAP	Yes	Yes	Yes	—
	Spoke-SDP	Yes	Yes	Yes	—
	Mesh-SDP	Yes	Yes	Yes	—
I-VPLS	—	—	—	—	No
	SAP	Yes	Yes	Yes	—
	Spoke-SDP	Yes	Yes	Yes	—
M-VPLS	—	—	—	—	No
	SAP	Yes	Yes	Yes	—
	Spoke-SDP	Yes	Yes	Yes	—
	Mesh-SDP	Yes	Yes	Yes	—
PBB Epipe	—	—	—	—	No
	SAP	Yes	Yes	Yes	—
	Spoke-SDP	Yes	Yes	Yes	—
Ipipe	—	—	—	—	No
	SAP	Yes	No	No	—
	Ethernet-Tunnel SAP	Yes	No	No	—
IES	—	—	—	—	No
	SAP	Yes	No	No	—
	Spoke-SDP (Interface)	Yes	No	No	—
	Subscriber Group-int SAP	Yes	No	No	—
VPRN	—	—	—	—	No
	SAP	Yes	No	No	—



Service	Ethernet connection	Down MEP	Up MEP	MIP	Virtual MEP
	Spoke-SDP (Interface)	Yes	No	No	—
	Subscriber Group-int SAP	Yes	No	No	—

Figure 40: MEP and MIP

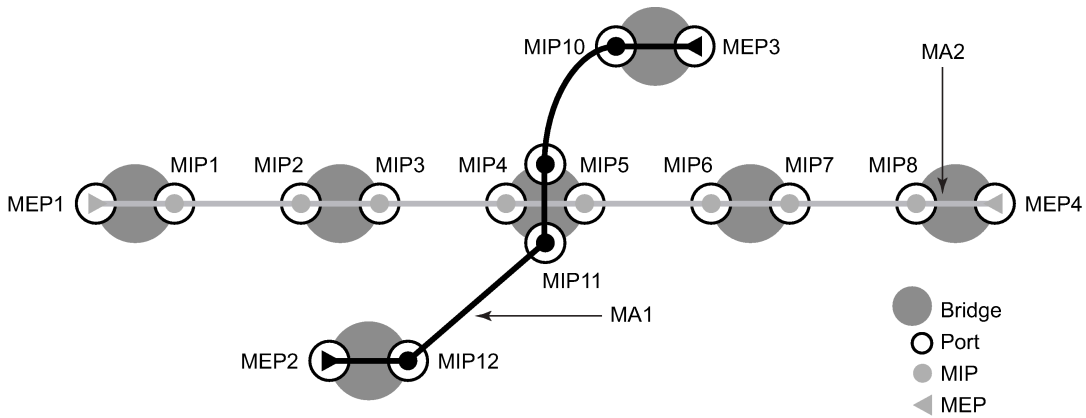
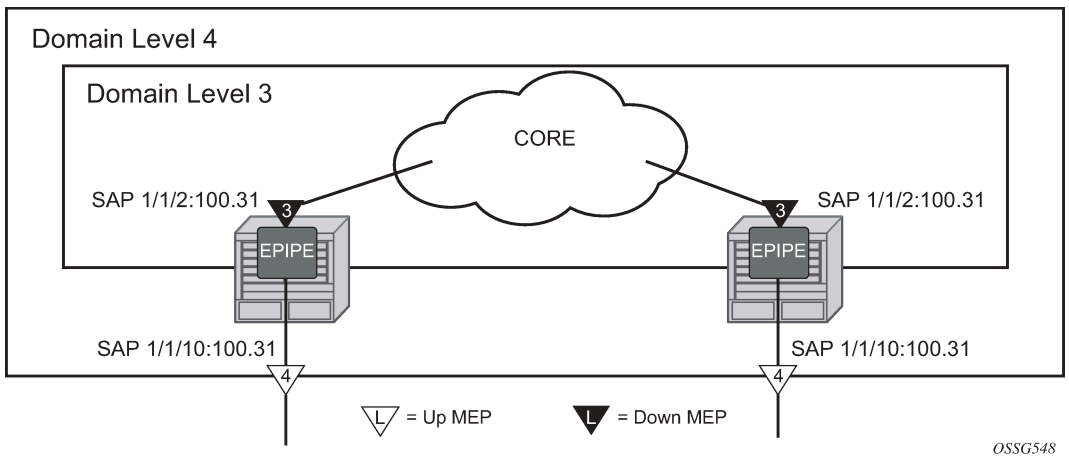


Figure 41: MEP creation illustrates the usage of an Epipe on two different nodes that are connected using ether SAP 1/1/2:100.31. The SAP 1/1/10:100.31 is an access port that is not used to connect the two nodes.

Figure 41: MEP creation



```

NODE1
config>eth-cfm# info
-----
domain 3 format none level 3
association 1 format icc-based name "03-0000000101"
bridge-identifier 100
exit
    
```

```

        exit
    exit
    domain 4 format none level 4
        association 1 format icc-based name "04-0000000102"
            bridge-identifier 100
        exit
    exit
exit
exit

config>service>epipe# info
-----
    sap 1/1/2:100.31 create
        eth-cfm
            mep 111 domain 3 association 1 direction down
                mac-address d0:0d:1e:00:01:11
                no shutdown
            exit
        exit
    exit
    sap 1/1/10:100.31 create
        eth-cfm
            mep 101 domain 4 association 1 direction up
                mac-address d0:0d:1e:00:01:01
                no shutdown
            exit
        exit
    exit
    no shutdown
-----

NODE 2
eth-cfm# info
-----
    domain 3 format none level 3
        association 1 format icc-based name "03-0000000101"
            bridge-identifier 100
        exit
    exit
    domain 4 format none level 4
        association 1 format icc-based name "04-0000000102"
            bridge-identifier 100
        exit
    exit
exit

config>service>epipe# info
-----
    sap 1/1/2:100.31 create
        eth-cfm
            mep 112 domain 3 association 1 direction down
                mac-address d0:0d:1e:00:01:12
                no shutdown
            exit
        exit
    exit
    sap 1/1/10:100.31 create
        eth-cfm
            mep 102 domain 4 association 1 direction up
                mac-address d0:0d:1e:00:01:02
                no shutdown
            exit
        exit
    exit

```

no shutdown

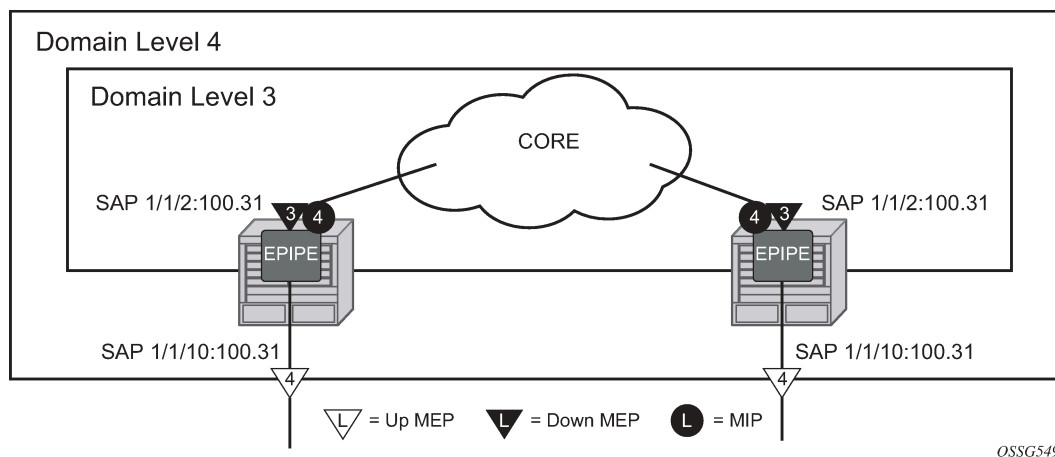
-----

Examining the configuration from NODE1, MEP 101 is configured with a direction of UP causing all ETH-CFM traffic originating from this MEP to generate into the switch fabric and out the mate SAP 1/1/2:100.31. MEP 111 uses the default direction of DOWN causing all ETH-CFM traffic that is generated from this MEP to send away from the fabric and only egress the SAP on which it is configured, SAP 1/1/2:100.31.

Further examination of the domain constructs reveal that the configuration properly uses domain nesting rules. In this case, the Level 3 domain is completely contained in a Level 4 domain.

[Figure 42: MIP creation example \(NODE1\)](#) illustrates the creation of an explicit MIP using the association MIP construct.

*Figure 42: MIP creation example (NODE1)*



```

NODE1
config>eth-cfm# info
-----
    domain 3 format none level 3
      association 1 format icc-based name "03-0000000101"
        bridge-identifier 100
        exit
      exit
    exit
  domain 4 format none level 4
    association 1 format icc-based name "04-0000000102"
      bridge-identifier 100
      exit
    exit
  association 2 format icc-based name "04-MIP00000102"
    bridge-identifier 100
    mhf-creation explicit
    exit
  exit
  exit
config>service>epipe# info
-----
  sap 1/1/2:100.31 create
  eth-cfm
    mep 111 domain 3 association 1 direction down
    mac-address d0:0d:1e:00:01:11

```

```

        no shutdown
        exit
    exit
exit
sap 1/1/10:100.31 create
    eth-cfm
        mep 101 domain 4 association 1 direction up
        mac-address d0:0d:1e:00:01:01
        no shutdown
    exit
    exit
exit
no shutdown
-----

NODE 2
eth-cfm# info
-----
    domain 3 format none level 3
        association 1 format icc-based name "03-0000000101"
        bridge-identifier 100
        exit
    exit
    domain 4 format none level 4
        association 1 format icc-based name "04-0000000102"
        bridge-identifier 100
        exit
    exit
    association 2 format icc-based name "04-MIP00000102"
        bridge-identifier 100
        mhf-creation explicit
        exit
    exit
exit
-----

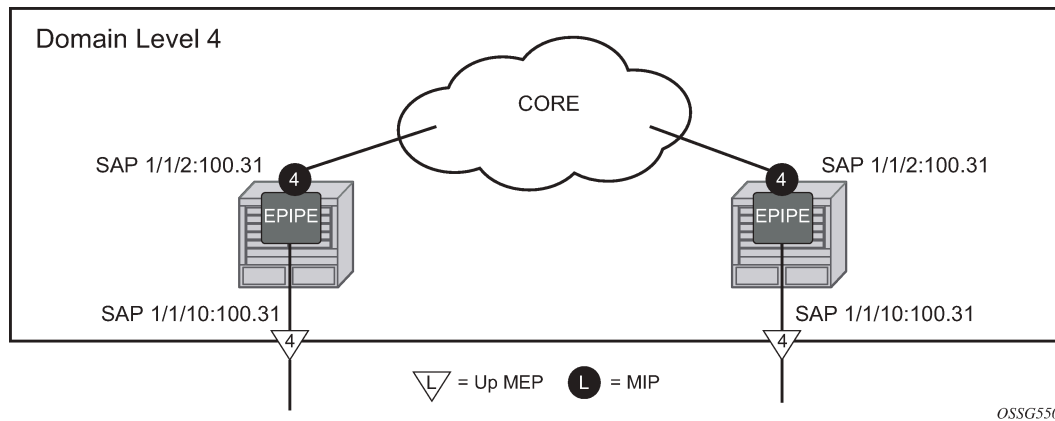
config>service>epipe# info
-----
    sap 1/1/2:100.31 create
        eth-cfm
            mep 112 domain 3 association 1 direction down
            mac-address d0:0d:1e:00:01:12
            no shutdown
        exit
    exit
    sap 1/1/10:100.31 create
        eth-cfm
            mep 102 domain 4 association 1 direction up
            mac-address d0:0d:1e:00:01:02
            no shutdown
        exit
    exit
    exit
no shutdown
-----

```

An addition of association 2 under domain 4 includes the **mhf-creation explicit** statement. This means that when the level 3 MEP is assigned to the SAP 1/1/2:100.31 using the definition in domain 3 association 1, creating the higher level MIP on the same SAP. Because a MIP does not have directionality "Both" sides are active. The service configuration and MEP configuration within the service did not change.

**Figure 43: MIP creation default** illustrates a simpler method that does not require the creation of the lower level MEP. The operator simply defines the association parameters and uses the **mhf-creation default** setting, then places the MIP on the SAP of their choice.

*Figure 43: MIP creation default*



NODE1:

```

config>eth-cfm# info
-----
  domain 4 format none level 4
    association 1 format icc-based name "04-0000000102"
      bridge-identifier 100
      exit
    exit
    association 2 format icc-based name "04-MIP00000102"
      bridge-identifier 100
      mhf-creation default
      exit
    exit
  exit
exit
-----

config>service>epipe# info
-----
  sap 1/1/2:100.31 create
    eth-cfm
      mip mac d0:0d:1e:01:01:01
    exit
  exit
  sap 1/1/10:100.31 create
    eth-cfm
      mep 101 domain 4 association 1 direction up
      mac-address d0:0d:1e:00:01:01
      no shutdown
    exit
  exit
  exit
  no shutdown
-----

```

NODE2:

```

config>eth-cfm# info
-----
    domain 4 format none level 4
      association 1 format icc-based name "04-0000000102"
        bridge-identifier 100
        exit
      exit
      association 2 format icc-based name "04-MIP0000102"
        bridge-identifier 100
        mhf-creation default
        exit
      exit
    exit
  exit
-----

config>service>epipe# info
-----
    sap 1/1/2:100.31 create
      eth-cfm
        mip mac d0:0d:1e:01:01:02
        exit
    exit
    sap 1/1/10:100.31 create
      eth-cfm
        mep 102 domain 4 association 1 direction up
        mac-address d0:0d:1e:00:01:02
        no shutdown
        exit
    exit
  exit
  no shutdown
-----

```

[Figure 44: MEP, MIP and MD levels](#) shows the detailed IEEE representation of MEPs, MIPs, levels and associations, using the standards defined icons.

SAPs support a comprehensive set of rules including wild cards to map packets to services. For example, a SAP mapping packets to a service with a port encapsulation of QinQ may choose to only look at the outer VLAN and wildcard the inner VLAN. SAP 1/1/1:100.\* would map all packets arriving on port 1/1/1 with an outer VLAN 100 and any inner VLAN to the service the SAP belongs to. These powerful abstractions extract inbound ETH-CFM PDUs only when there is an exact match to the SAP construct. In the case of the example when then an ETH-CFM PDU arrives on port 1/1/1 with a single VLAN with a value of 100 followed immediately with e-type (0x8902 ETH-CFM). Furthermore, the generation of the ETH-CFM PDUs that egress this specific SAP are sent with only a single tag of 100. The primary VLAN is required if the operator needs to extract ETH-CFM PDUs or generate ETH-CFM PDUs on wildcard SAPs and the offset includes an additional VLAN that was not part of the SAP configuration.

[Table 13: Extraction comparison with primary VLAN](#) shows how packets that would normally bypass the ETH-CFM extraction would be extracted when the primary VLAN is configured. This assumes that the processing rules for MEPs and MIPs is met, E-type 0x8902, Levels and OpCodes.

Table 13: Extraction comparison with primary VLAN

Port encapsulation	E-type	Ingress tags	Ingress SAP	No primary VLAN ETH-CFM extraction		With primary VLAN (10) ETH-CFM extraction	
				MEP	MIP	MEP	MIP
—	—	—	—	MEP	MIP	MEP	MIP
Dot1q	0x8902	10	x/y/z:*	No	No	Yes	Yes
Dot1q	0x8902	10.10	x/y/z:10	No	No	Yes	Yes
QinQ	0x8902	10.10	x/y/z:10.*	No	No	Yes	Yes
QinQ (Default Behavior)	0x8902	10.10	x/y/z:10.0	No	No	Yes	Yes
Null	0x8902	10	x/y/z	No	No	Yes	Yes

The mapping of the service data remains unchanged. The primary VLAN function allows for one additional VLAN offset beyond the SAP configuration, up to a maximum of two VLANs in the frame. If a fully qualified SAP specifies two VLANs (SAP 1/1/1:10.10) and a primary VLAN of 12 is configured for the MEP there is no extraction of ETH-CFM for packets arriving tagged 10.10.12. That exceeds the maximum of two tags.

The mapping or service data based on SAPs has not changed. ETH-CFM MPs functionality remains SAP specific. In instances where as service includes a specific SAP with a specified VLAN (1/1/1:50) and a wildcard SAP on the same port (1/1/1:\*) it is important to understand how the ETH-CFM packets are handled. Any ETH-CFM packet with etype 0x8902 arriving with a single tag or 50 would be mapped to a classic MEP configured under SAP 1/1/1:50. Any packet arriving with an outer VLAN of 50 and second VLAN of 10 would be extracted by the 1/1/1:50 SAP and would require a primary VLAN enabled MEP with a value of 10, assuming the operator would like to extract the ETH-CFM PDU of course. An inbound packet on 1/1/1 with an outer VLAN tag of 10 would be mapped to the SAP 1/1/1:\*. If ETH-CFM extraction is required under SAP 1/1/1:\* a primary VLAN enabled MEP with a value of 10 would be required.

The packet that is generated from a MEP or MIP with the primary VLAN enabled is include that VLAN. The SAP encapsulates the primary VLAN using the SAP encapsulation.

Primary VLAN support includes UP MEPs, DOWN MEPs and MIPs on Ethernet SAPs, including LAG, as well as SDP bindings for Epipe and VPLS services. Classic MEPs, those without a primary VLAN enabled, and a primary VLAN enabled MEPs can coexist under the same SAP or SDP binding. Classic MIPs and primary VLAN-enabled MIPs may also coexist. The enforcement of a single classic MIP per SAP or SDP binding continues to be enforced. However, the operator may configure multiple primary VLAN-enabled MIPs on the same SAP or SDP binding. MIPs in the primary VLAN space must include the **mhf-creation static** configuration under the association and must also include the specific VLAN on the MIP creation statement under the SAP. The **no** version of the **mip** command must include the entire statement including the VLAN information.

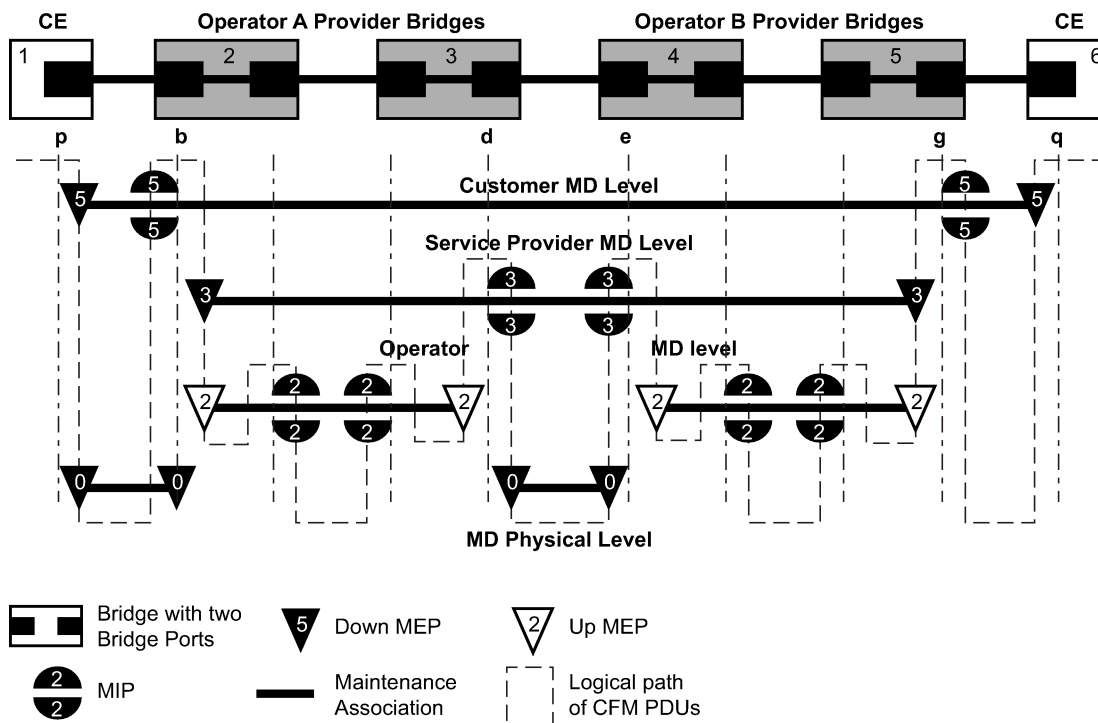
The eight MD Levels (0 to 7) are specific to context in which the Management Point (MP) is configured. This means the classic MPs have a discrete set of the levels from the primary VLAN enabled space. Each primary VLAN space has its own eight Level MD space for the specified primary VLAN. Consideration must be extended before allowing overlapping levels between customers and operators should the operator be provision a customer facing MP, like a MIP on a UNI. CPU Protection extensions for ETH-CFM are VLAN

unaware and based on MD Level and the OpCode. Any configured rates are applied to the Level and OpCode as a group.

There are two configuration steps to enable the primary VLAN. Under the bridging instance, contained within the association context (`config>eth-cfm>domain>assoc>bridge`), the VLAN information must be configured. Until this is enabled using the `primary-vlan-enable` option as part of the MEP creation step or the MIP statement (`config>service>...>{sap | mesh-sdp | spoke-sdp}>eth-cfm`) the VLAN specified under the bridging instance remains inactive. This is to ensure backward interoperability.

Primary VLAN functions require an FP2-based card or better. Primary VLAN is not supported for vpls-sap-templates, sub-second CCM intervals, or vMEPs.

Figure 44: MEP, MIP and MD levels



An operator may see the following INFO message (during configuration reload), or MINOR (error) message (during configuration creation) when upgrading to 11.0r4 or later if two MEPs are in a previously undetected conflicting configuration. The messaging is an indication that a MEP, the one stated in the message using format (`domain md-index/association ma-index/mep mep-id`), is already configured and has allocated that context. During a reload (INFO) a MEP that encounters this condition is created but its state machine is disabled. If the MINOR error occurs during a configuration creation this MEP fails the creation step. The indicated MEP must be correctly re-configured.

```
INFO: ETH_CFM #1341 Unsupported MA ccm-interval for this MEP - MEP 1/112/
21 conflicts with sub-second config on this MA
MINOR: ETH_CFM #1341 Unsupported MA ccm-interval for this MEP - MEP 1/112/
21 conflicts with sub-second config on this MA
```

Service data arriving at an ingress SAP performs several parsing operations to map the packet to the service as well as to VLAN functions. VLAN functions include determining the service-delineated VLANs based on the ingress configuration. Locally-generated CFM packets are unaware of the ingress



VLAN functions. This may lead to service data and CFM data tagging alignment issues when the egress connection is a binding. For example, if the SDP is configured with **vc-type vlan** and the binding referencing the SDP does not specify the VLAN tag with the optional **vlan-vc-tag vlan-id** configuration, the service data crossing the service and the locally-generated CFM packets can use different VLAN tags. A problem can occur if this VLAN is significant to the peer. Similarly, an EVPN service cannot specify the **vlan-vc-tag vlan-id** to be used on the binding.

The optional **cfm-vlan-tag <qtag1[.<qtag2>]>** command used for MEP and MIP configurations supports the alignment of service data and the locally generated CFM packet VLAN tags for bindings that require matching VLAN tags. The qtag configuration should typically match the ingress SAP configuration. For example, a SAP that is configured with an **enacp-type qinq** and the associated SAP of 100.\* should consider using the **cfm-vlan-tag 100** configuration option under the MEP or MIP, when a situation describing misalignment of VLAN tags is encountered.

The **cfm-vlan-tag <qtag1[.<qtag2>]>** command option is supported for VPLS and Epipe services only.

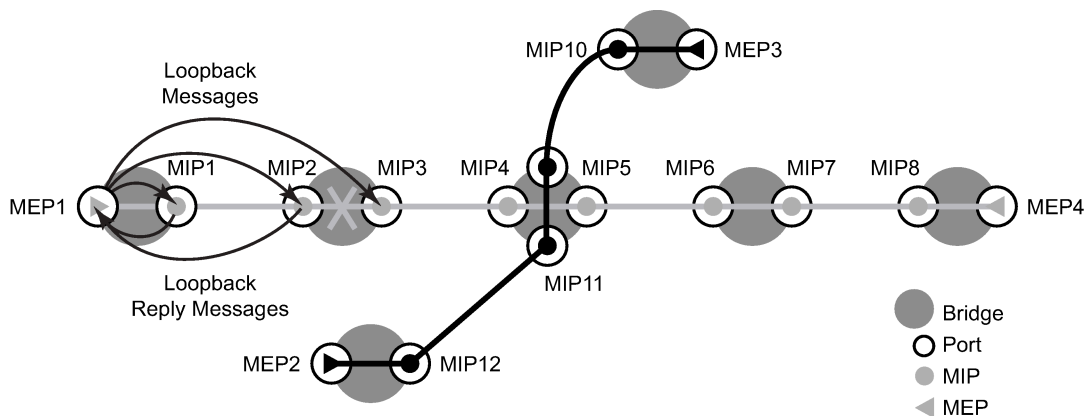
The **cfm-vlan-tag <qtag1[.<qtag2>]>** command option is not supported for the following cases:

- when the configuration requires the CFM to include more than two VLAN tags, which may cause a configuration error; invalid configurations include a MEP or MIP configured with a **primary-vlan-enable vlan-id** and both tags specified of the **cfm-vlan-tag <qtag1[.<qtag2>]>**
- when the MIP creation does not include the MIP configuration statement under the service, specifically, the default behavior MIP that is created solely based on the associated MHF creation
- when the context is **config>service>template vpls-sap-template**
- for G.8031 (ETH-tunnel) and G.8032 (ETH-ring)
- for a MIP on a PW-SAP

### 3.4.2 Loopback

A loopback message is generated by an MEP to its peer MEP or a MIP (Figure 45: CFM loopback). The functions are similar to an IP ping to verify Ethernet connectivity between the nodes.

Figure 45: CFM loopback



The following loopback-related functions are supported:

- Loopback message functionality on an MEP or MIP can be enabled or disabled.



Because ETH-LB does not support standard timestamps, no indication of delay is produced as these times are not representative of network delay.

By default, if no interval is included in the command, the default is back to back LBM transmissions. The maximum count for such a test is 5.

### 3.4.3 Loopback multicast

Multicast loopback also supports the new intervals (see [Loopback](#)). However, the operator must be careful when using this approach. Every MEP in the association responds to this request. This means an exponential impact on system resources for large scale tests. If the multicast option is used and there with an interval of 1 (100ms) and there are 50 MEPs in the association, this results in a 50 times increase in the receive rate (500pps) compared to a unicast approach. Multicast displays are not be updated until the test is completed. There is no packet loss percentage calculated for multicast loopback commands.

This on demand operation tool is used to quickly check the reachability of all MEPs within an Association. A multicast address can be coded as the destination of an **oam eth-cm loopback** command. The specific class 1 multicast MAC address or the keyword "multicast" can be used as the destination for the loopback command. The class 1 ETH-CFM multicast address is in the format 01:80:C2:00:00:3x (where x = 0 - 7 and is the number of the domain level for the source MEP). When the "multicast" option is used, the class 1 multicast destination is built according to the local MEP level initiating the test.

Remote MEPs that receive the multicast loopback message, configured at the equivalent level, are terminated and process the multicast loopback message by responding with the appropriate unicast loopback response (ETH-LBR). Regardless of whether a multicast or unicast ETH-LBM is used, there is no provision in the standard LBR PDU to carry the MEP-ID of the responder. This means only the remote MEP MAC Address is reported and subsequently displayed. MIPs do not extract a multicast LBM request. The LBM multicast is transparent to the MIP.

MEP loopback stats are not updated as a result of this test being run. That means the received, out-of-order and bad-msdu counts are not affected by multicast loopback tests. The multicast loopback command is meant to provide immediate connectivity troubleshooting feedback for remote MEP reachability only.

```
oam eth-cfm loopback multicast mep 28 domain 14 association 2 interval 1 send-
count 100
Eth-Cfm Loopback Test Initiated: Mac-Address: multicast, out service: 5
```

MAC Address	Receive Order															
00:00:00:00:00:30	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
4	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97
98	99	100														
00:00:00:00:00:32	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
4	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97
98	99	100														

```
Sent 100 multicast packets, received 200 packets
```

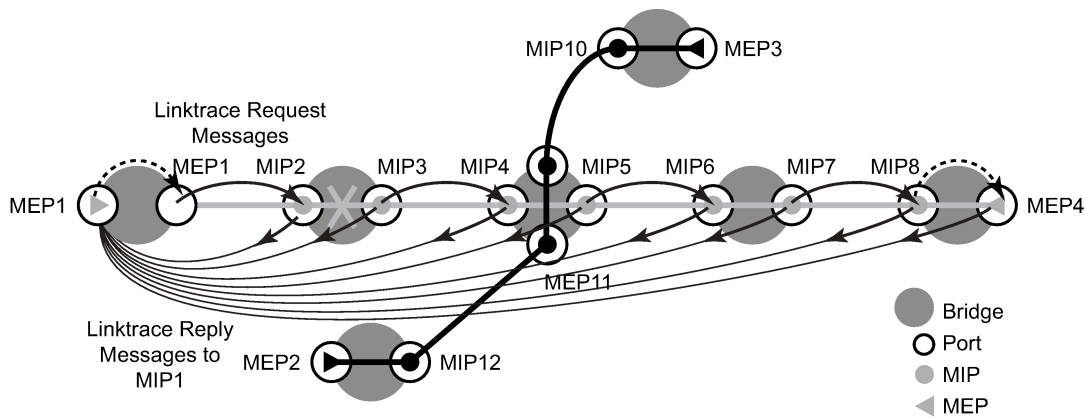
### 3.4.4 Linktrace

A linktrace message is originated by an MEP and targeted to a peer MEP in the same MA and within the same MD level (Figure 46: CFM linktrace). Linktrace traces a specific MAC address through the service. The peer MEP responds with a linktrace reply message after successful inspection of the linktrace message. The MIPs along the path also process the linktrace message and respond with linktrace replies to the originating MEP if the received linktrace message that has a TTL greater than 1 and forward the linktrace message if a look up of the target MAC address in the Layer 2 FDB is successful. The originating MEP shall expect to receive multiple linktrace replies and from processing the linktrace replies, it can put together the route to the target bridge.

A traced MAC address is carried in the payload of the linktrace message, the target MAC. Each MIP and MEP receiving the linktrace message checks whether it has learned the target MAC address. To use linktrace the target MAC address must have been learned by the nodes in the network. If so, a linktrace message is sent back to the originating MEP. Also, a MIP forwards the linktrace message out of the port where the target MAC address was learned.

The linktrace message itself has a multicast destination address. On a broadcast LAN, it can be received by multiple nodes connected to that LAN. But, at most, one node sends a reply.

Figure 46: CFM linktrace



Fig\_13

The following linktrace related functions are supported:

- MEP supports generating linktrace messages and responding with linktrace reply messages. The ETH-LT PDU format does not allow a MEP to have more than a single active ETH-LT session.
- MIP supports responding to linktrace messages with linktrace reply messages when encoded TTL is greater than 1, and forward the linktrace messages accordingly if a lookup of the target MAC address in the Layer 2 FDB is successful.
- Displays linktrace test results on the originating MEP.
- SenderID TLV may optionally be configured to carry the ChassisID. When configured, this information is included in LTM and LTR messages.
  - Only the ChassisID portion of the TLV is included.
  - The Management Domain and Management Address fields are not supported on transmission.

- THE LBM message includes the SenderID TLV that is configure on the launch point. The LBR message includes the SenderID TLV information from the reflector (MIP or MEP) if it is supported.
- Supported for both service (id-permission) and facility MEPs (facility-id-permission).
- Supported for both MEP and MIP.

The following output includes the SenderID TLV contents if it is included in the LBR.

```
oam eth-cfm linktrace 00:00:00:00:00:30 mep 28 domain 14 association 2
Index Ingress Mac          Egress Mac          Relay      Action
-----
1      00:00:00:00:00:00      00:00:00:00:00:30  n/a       terminate
SenderId TLV: ChassisId (local)
              access-012-west
-----
No more responses received in the last 6 seconds.
```

### 3.4.5 Continuity Check (CC)

A Continuity Check Message (CCM) is a multicast frame that is generated by a MEP and multicast to all other MEPs in the same MA. The CCM does not require a reply message. To identify faults, the receiving MEP maintains an internal list of remote MEPs it should be receiving CCM messages from.

This list is based off of the remote-mepid configuration within the association the MEP is created in. When the local MEP does not receive a CCM from one of the configured remote MEPs within a pre-configured period, the local MEP raises an alarm.

Figure 47: CFM CC

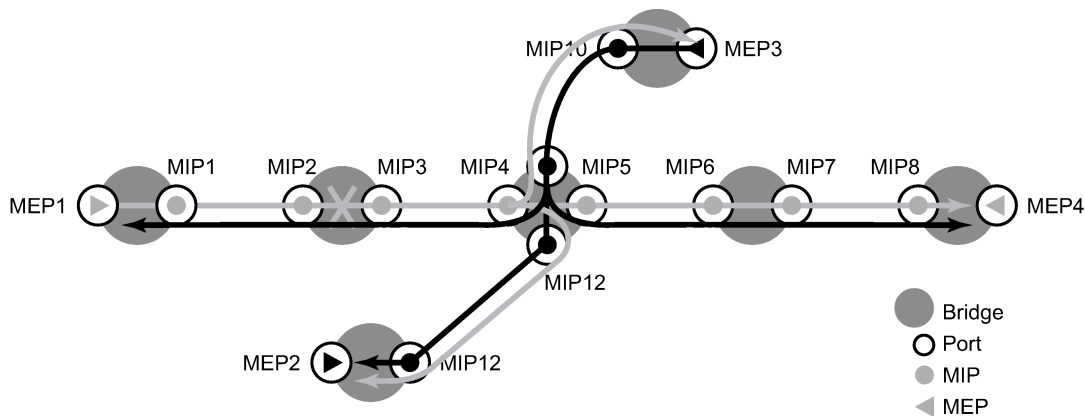
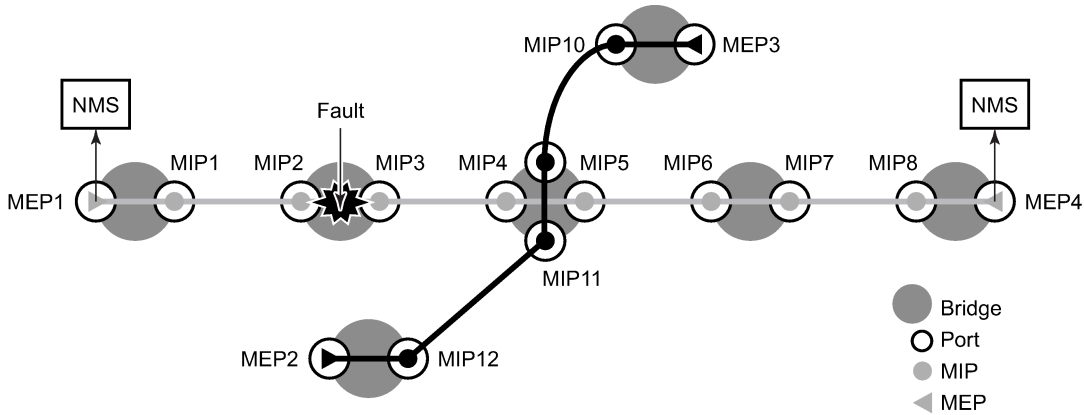
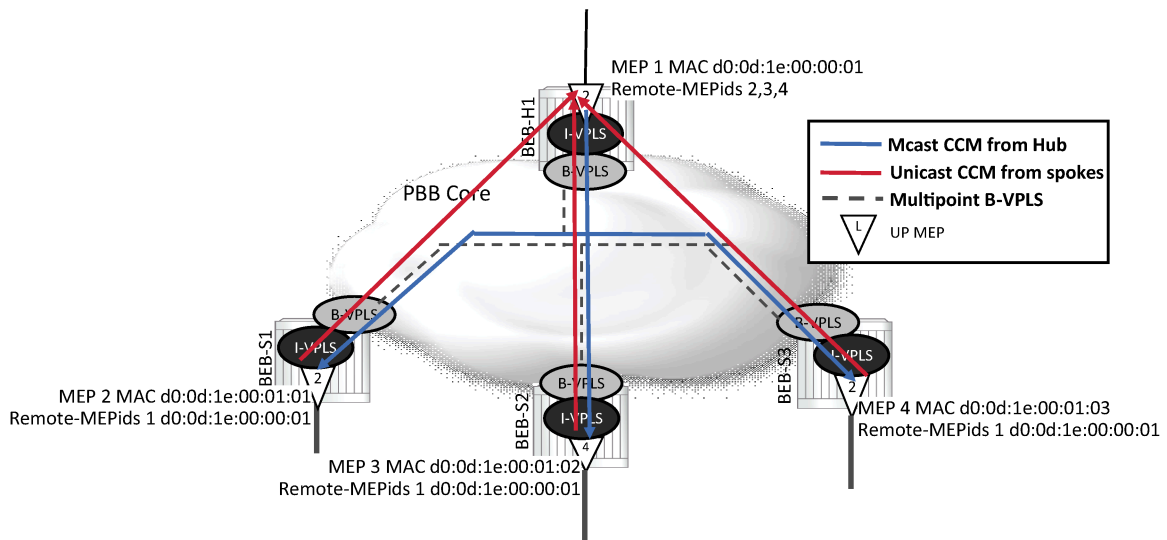


Figure 48: CFM CC failure scenario



An MEP may be configured to generate ETH-CC packet using a unicast destination Layer 2 MAC address. This may help reduce the overhead in some operational models where Down MEPs per peer are not available. For example, mapping an I-VPLS to a PBB core where a hub is responsible for multiple spokes is one of the applicable models. When ETH-CFM packets are generated from an I-context toward a remote I-context, the packets traverse the B-VPLS context. Because many B-contexts are multipoint, any broadcast, unknown or multicast packet is flooded to all appropriate nodes in the B-context. When ETH-CC multicast packets are generated, all the I-VPLS contexts in the association must be configured with all the appropriate remote MEPids. If direct spoke to spoke connectivity is not part of the validation requirement, the operational complexity can be reduced by configuring unicast DA addressing on the "spokes" and continuing to use multicast CCM from the "hub". When the unicast MAC is learned in the forwarding database, traffic is scoped to a single node.

Figure 49: Unicast CCM in hub and spoke environments



Defect condition, reception, and processing remains unchanged for both hub and spokes. When an ETH-CC defect condition is raised on the hub or spoke, the appropriate defect condition is set and distributed throughout the association from the multicasting MEP. For example, should a spoke raise a defect condition or timeout, the hub sets the RDI bit in the multicast ETH-CC packet which is received on all

spokes. Any local hub MEP defect condition continues to be propagated in the multicast ETH-CC packet. Defect conditions are cleared as per normal behavior.

The forwarding plane must be considered before deploying this type of ETH-CC model. A unicast packet is handled as unknown when the destination MAC does not exist in local forwarding table. If a unicast ETH-CC packet is flooded in a multipoint context, it reaches all the appropriate I-contexts. This causes the spoke MEPs to raise the "DefErrorCCM" condition because an ETH-CC packet was received from a MEP that has not been configured as part of the receiving MEPs database.

The remote unicast MAC address must be configured and is not automatically learned. A MEP cannot send both unicast and multicast ETH-CC packets. Unicast ETH-CC is only applicable to a local association with a single configured remote peer. There is no validation of MAC addresses for ETH-CC packets. The configured unicast destination MAC address of the peer MEP only replaces the multicast class 1 destination MAC address with a unicast destination.

Unicast CCM is not supported on any MEPs that are configured with sub second CCM-intervals.

The following functions are supported:

- Enable and disable CC for an MEP
- Configure and delete the MEP entries in the CC MEP monitoring database manually. It is only required to provision remote MEPs. Local MEPs shall be automatically put into the database when they are created.
- CCM transmit interval: 10ms, 100ms, 1s, 10s 60s, 600s. Default: 10s. Interval support is platform dependent. When configuring MEPs with sub-second CCM intervals, bandwidth consumption must be taken into consideration. Each CCM PDU is approximately 100 bytes (800 bits). Taken individually, this is a small value. However, the bandwidth consumption increases rapidly as multiple MEPs are configured with 10ms timers, 100 packets per second.

The following section describes some basic hierarchical considerations and the software requirements and configurations that need to be met when considering sub-second enabled MEPs.

- Down MEPs only
- Single peer only
- Any MD Level
  - As long as lower MD level MEPs are not CCM or ETH-APS enabled
    - G.8031 Ethernet-Tunnels enables OpCode39 Linear APS
    - G.8032 Ethernet-Rings enables OpCode 40 Ring APS
  - As long as lower MD levels MEPs are not receiving ETH-CCM or ETH-APS PDUs, even if they not locally enabled or configured to do so
    - The reception of the lower MD level ETH-CCM and ETH-APS PDUs are processed by the sub second CCM enabled MEP, regardless of MD Level
    - All other ETH-CFM PDUs are handled by the MEP at the MD level matching the PDU that has arrived, assuming one has been configured
- Service MEPs (excluding primary VLAN MEPs)
  - Ethernet SAPs configured on Port with any Ethernet Encapsulation (null, dot1q or QinQ)
- Facility MEPs
  - Ethernet Port Based MEPs
  - Ethernet LAG Based MEPs

- Ethernet QinQ Tunnel based MEPs (LAG+VLAN, PORT+VLAN)
- Base Router IP Interfaces
- Service MEPs and Facility MEPs can simultaneously execute sub second CCM enabled MEPs as these are considered different MEP families.
- General processing rules for Service MEPs and Facility MEPs must be met regardless of the CCM interval. These are included here because of the impact misunderstanding could have on the CCM extraction.
  - All the above rules apply
  - MD level hierarchy must be ensured across different families
  - Facility MEPs are the first processing routine for ETH-CFM PDUs
  - VLAN encapsulation uniqueness must exist when processing the ETH-CFM PDU across the two families
 

Unique Example: An Ethernet Port Based Facility Down MEP configured on port 1/1/1 and Service Down MEP SAP 1/1/1:100 (dot1q encaps) are unique

Conflict Example: An Ethernet Port Based Facility Down MEP configured on port 1/1/1 and Service Down MEP SAP 1/1/1 (null encaps) are in conflict and cannot coexist. All ETH-CFM PDUs arrive untagged and the Facility MEP takes precedence.
- G.8031 (Ethernet-Tunnels) support both sub second and 1 second CCM intervals and optionally no CCM. When the MEP is created on a G.8031 Ethernet-Tunnel no other MEP that is any way connected to the G.8031 Ethernet-Tunnel can execute sub second CCM intervals. Facility MEPs are not supported in conjunction with G.8031 (Ethernet-Tunnel MEPs)
- G.8032 (Ethernet-Ring) support both sub second and 1 second CCM intervals and optionally no CCM. Facility MEPs are supported in combination with G.8032 MEPs. However, facility MEPs and G.8032 MEPs cannot both execute sub second CCM where the infrastructure is shared. If the operator configures this combination the last updated sub second MEP overwrites the previous sub second MEP and interrupt the previous configured MEP causing a defRemoteCCM condition.
- The size of the CCM PDU may be increased by configuring the optional Data TLV. This is accomplished by configuring the **ccm-padding-size** command under the specific MEP. The configured value represents the total length of the Data TLV that is included with the other CCM PDU informational elements. The **no** form of this command removes the optional Data TLV from the CCM PDU. The operator must consider a CCM PDU is 83 byte size in length (75 base elements plus 8 bytes for port status and interface status). If the size of the optional TLV combined with the size of the CCM PDU exceeds 1500 bytes the packet is dropped if the MTU is 1518/1522.
- CCM declares a defect when:
  - it stops hearing from one of the remote MEPs for 3.5 times CC interval
  - it hears from a MEP with a LOWER MD level
  - it hears from a MEP that is not part of the local MEPs MA
  - it hears from a MEP that is in the same MA but not in the configured MEP list
  - it hears from a MEP in the same MA with the same MEP ID as the receiving MEP
  - the CC interval of the remote MEP does not match the local configured CC interval
  - the remote MEP is declaring a fault



- An alarm is raised and a trap is sent if the defect is greater than or equal to the configured low-priority-defect value.
- Remote Defect Indication (RDI) is supported but by default is not recognized as a defect condition because the low-priority-defect setting default does not include RDI.
- SenderID TLV may optionally be configured to carry the ChassisID. When configured, this information is included in CCM messages.
  - Only the ChassisID portion of the TLV is included.
  - The Management Domain and Management Address fields are not supported on transmission.
  - SenderID TLV is not supported with sub second CCM enabled MEPs.
  - Supported for both service (id-permission) and facility MEPs (facility-id-permission).
- Alarm notification alarm and reset times are configurable under the MEP. By default, the alarm notification times are set to zero, which means the behavior is immediate logging and resetting. When the value is zero and a previous higher level alarm is reset, if a lower level alarm exists, and is above the low-priority defect, a log event is created. However, when either of the alarm notification timers are non-zero and a lower priority alarm exists, it is not logged.
  - Alarm (fng-alarm-time) delays the generation of the log event by the value configured. The alarm must be present for this amount of time before the log event is created. This is for only log event purposes.
  - Reset (fng-reset-time) is the amount of time the alarm must be absent before it is cleared.

The optional **ccm-tlv-ignore** command ignores the reception of interface-status and port-status TLVs in the ETH-CCM PDU on Facility MEPs (port, LAG, QinQ, tunnel and router). No processing is performed on the ignored ETH-CCM TLVs values.

Any TLV that is ignored is reported as *absent* for that remote peer and the values in the TLV do not have an impact on the ETH-CFM state machine. This the same behavior as if the remote MEP never included the ignored TLVs in the ETH-CCM PDU. If the TLV is not properly formed, the CCM PDU fails the packet parsing process, which causes it to be discarded and a defect condition is raised.

There are various display commands that are available to show the status of the MEP and the list of remote peers.

### 3.4.6 CC remote peer auto discovery

As specified in the section "Continuity Checking (CC)," all remote MEP-IDs must be configured under the association using the **remote-mepid** command to accept them as peers. When a CCM is received from a MEP-ID that has not been configured, the "unexpected MEP" causes the defErrorCCM condition to be raised. The defErrorCCM is raised for all invalid CC reception conditions.

The auto-mep-discovery option allows for the automatic adding of remote MEP-IDs contained in the received CCM. When learned, the automatically discovered MEP behave the same as a manually configured entry. This includes the handling and reporting of defect conditions. For example, if an auto discovered MEP is deleted from its host node, it experiences the standard timeout on the node which auto discovered it.

When this function is enabled, the "unexpected MEP" condition no longer exists. That is because all MEPs are accepted as peers and automatically added to the MEP database upon reception. There is an exception to this statement. If the maintenance association has reached its maximum MEP count, and no

new MEPs can be added, the “unexpected MEP” condition raises the defErrorCCM defect condition. This is because the MEP was not added to the association and the remote MEP is still transmitting CCM.

The **clear eth-cfm auto-discovered-meps** [*mep-id*] **domain** *md-index* **association** *ma-index* is available to remove auto discovered MEPs from the association. When the optional *mep-id* is included as part of the clear command, only that specific MEP-ID within the domain and association is cleared. If the optional *mep-id* is omitted when the clear command is issued, all auto discovered MEPs that match the domain and association are cleared. The **clear** command is only applicable to auto-discovered MEPs.

If there is a failure to add a MEP to the MEP database and the action was manual addition using the “remote-mepid” configuration statement, the error “MINOR: ETH\_CFM #1203 Reached maximum number of local and remote endpoints configured for this association” is produced. When failure to add a MEP to the database through an auto discovery, no event is created. The CCM Last Failure indicator tracks the last CCM error condition. The decode can be viewed using the **show eth-cfm mep** *mep-id* **domain** *md-index* **association** *ma-index* command. An association may include both the manual addition of remote peers using the remote-mepid and the auto-mep-discovery option.

The all-remote-mepid display includes an additional column AD to indicate where a MEP has been auto discovered, using the indicator T.

Auto discovered MEPs do not survive a system reboot. These are not permanent additions to the MEP database and are not reloaded after a reboot. The entries are relearned when the CCM is received. Auto discovered MEPs can be changed to manually created entries simply by adding the appropriate remote-mepid statement to the correct association. At that point, the MEP is no longer considered auto discovered and can no longer be cleared.

If a remote-mepid statement is removed from the association context and auto-mep-discovery is configured and a CC message arrives from that remote MEP, it is added to the MEP database, this time as an auto discovered MEP.

The individual MEP database for an association must not exceed the maximum number of MEPs allowed. A MEP database consists of all local MEPs plus all configured remote-mepids and all auto-discovered MEPs. If the number of MEPs in the association has reached capacity, no new MEPs may be added. The number of MEPs must be brought below the maximum value before MEPs can be added. Also, the number of MEPs across all MEP databases must not exceed the system maximum. The number of MEPs supported per association and the total number of MEPs across all associations is platform dependent.

### 3.4.7 ETH-CFM grace overview

ETH-CFM grace is an indication that MEPs on a node undergoing a maintenance operation may be expected to be unable to transmit or receive ETH-CC PDUs, failing to satisfy the peers requirements. Without the use of a supporting grace function, CCM-enabled MEPs time out after an interval of  $3.5 \times \text{ccm-interval}$ . During planned maintenance operations, the use of grace can extend the timeout condition to a longer interval.

The Ethernet CFM system-wide configuration **eth-cfm>system>[no] grace-tx-enable** command controls the transmission of ETH-CFM grace. The ETH-CFM grace function is enabled by the Soft Reset notification by default. The ETH-CFM grace function determines the individual MEP actions based on their configured parameters.

To transmit a grace PDU, the MEP must be administratively enabled and ETH-CC must also be enabled. The ETH-CC interval is ignored. Grace transmission uses the class 1 DA, with the last nibble (4 bits) indicating the domain level, for all grace-enabled MEPs. When a grace event occurs, all MEPs on a node that are configured for grace actively participate in the grace function until the grace event has completed. When a soft reset occurs, ETH-CFM does not determine which peers are directly affected by a soft reset

of a specific IOM or line card. This means that all MEPs enter a grace state, regardless of their location on the local node.

The grace process prevents the local MEP from presenting a new timeout condition, and prevents its peer, also supporting a complementary grace process, from declaring a new timeout defect (DefRemoteCCM). Other defects, unrelated to timeout conditions, are processed as during normal operation. This includes the setting, transmission, and reception processing of the RDI flag in the CCM PDU. Because the timeout condition has been prevented, it can be assumed that the RDI is caused by some other unrelated CCM defect condition. Entering the grace period does not clear existing defect conditions, and any defect condition that exists at the start of the grace period is maintained and cleared using normal operation.

Two approaches are supported for ETH-CFM grace:

- [ETH-VSM grace \(Nokia SR OS vendor-specific\)](#)
- [ITU-T Y.1731 ETH-ED](#)

Both approaches use the same triggering infrastructure but have unique PDU formats and processing behaviors. Only one grace transmission function can be active under an individual MEP. MEPs can be configured to receive and process both grace PDU formats. If a MEP receives both types of grace PDUs, the last grace PDU received becomes the authority for the grace period, using its procedures. If the operator needs to clear a grace window or expected defect window on a receiving peer, the appropriate authoritative reception function can be disabled.

Active AIS server transmissions include a vendor-specific TLV that instructs the client to extend the timeout of AIS during times of grace. When the grace period is completed, the server MEP removes the TLV and the client reverts to standard timeout processing based on the interval in the AIS PDU.

### 3.4.7.1 ETH-VSM grace (Nokia SR OS vendor-specific)

The ETH-VSM Multicast Class 1 DA announcement includes the start of a grace period, the new remote timeout value of 90 s, and the completion of the grace process.

At the start of the maintenance operation, a burst of three packets is sent over a 3-second window to reduce the chance that a remote peer may miss the grace announcement. Following the initial burst, evenly-spaced ETH-VSM packets are sent at intervals of one third of the ETH-VSM grace window; this means that the ETH-VSM packets are sent every 30 seconds to all appropriate remote peers. Reception of an ETH-VSM grace packet refreshes the timeout calculation. The local node that is undergoing the maintenance operation also delays the CCM timeout of the local MEP during the grace window using the announced ETH-VSM interval. MEPs restart their timeout countdown when any ETH-CC PDU is received.

At the end of the maintenance operation, there is a burst of three ETH-VSM grace packets to signal that the maintenance operation has been completed. After the first of these packets has been received, the receiving peer transitions back to the ETH-CCM message and associated interval as the indication for the remote timeout ( $3.5 \times \text{ccm-interval} + \text{hold}$  (where applicable)).

CCM packets continue to be sent during this process, but loss of the CCM packets during the advertised grace window do not affect the peer timeout. The only change to the CCM processing is the timeout value used during the grace operation. During the operation, the value that is announced as part of the ETH-VSM packet is used. If the grace value is lower than the configured CCM interval standard timeout computation ( $3.5 \times \text{ccm-interval} + \text{hold}$  (where applicable)), the grace value is not installed as the new timeout metric.

This is a value-added function that is applicable only to nodes that implement support for Nokia's approach for announcing grace using ETH-VSM. This pre-dates the introduction of the ITU-T Y.1371 Ethernet-

Expected Defect (ETH-ED) standard. As specified in the standards, when a node does not support a specific optional function such as ETH-VSM, the message is ignored and no processing is performed.

The ETH-VSM function is enabled by default for reception and transmission. The per-MEP configuration statements under the **grace>eth-vsm-grace** context can affect the transmission, reception, and processing of the ETH-VSM grace function.

### 3.4.7.2 ITU-T Y.1731 ETH-ED

The ETH-ED PDU is used to announce the expected defect window to peer MEPs. The peer MEPs uses the expected defect window value to prevent ETH-CC timeout (DefRemoteCCM) conditions for the announcing MEP. The MEP announcing ETH-ED does not time out any remote peers during the expected defect window. The expected defect window is not a configurable value.

At the start of the operation, a burst of three packets are sent over a 3-second window to reduce the chance that a remote peer may miss the expected defect window announcement.

It is possible to restrict the value that is installed for the expected defect timer by configuring the **max-rx-defect-window** command for the receiving MEP. A comparison is used to determine the expected defect timer to be installed during grace. Either the lower of the received expected defect timer values in the ETH-ED PDU or the configured maximum is installed if they are larger than the standard computation for ETH-CC timeout. The **no max-rx-defect-window** command is configured by default; therefore, the maximum received expected defect window is disabled, and it is not considered in determining the installed expected defect timer.

Subsequent ETH-ED packets are only transmitted at the completion of the Soft Rest function that triggered the grace function. The three-packet burst at the completion of the Soft Reset function contains an expected defect window size of 5 seconds. Receiving peers should use this new advertisement to reset the expected window to 5 seconds.

The termination of the grace window occurs when the expected defect window timer reaches zero, or when the receive function is manually disabled.

### 3.4.8 CCM hold timers

In some cases, the requirement exists to prevent a MEP from entering the defRemoteCCM defect, remote peer timeout, for more time than the standard 3.5 times the **ccm-interval**. Both the IEEE 802.1ag standard and ITU-T Y.1731 recommendation provide a non-configurable 3.5 times the CCM interval to determine a peer time out. However, when sub-second CCM timers (10 ms/100 ms) are enabled, the carrier may want to provide additional time for different network segments to converge before declaring a peer lost because of a timeout. To maintain compliance with the specifications, the **ccm-hold-timer down delay-down** option artificially increases the amount of time it takes for a MEP to enter a failed state if the peer times out. This timer is only additive to CCM timeout conditions. All other CCM defect conditions, like defMACStatus, defXconCCM, and so on, maintain their existing behavior of transitioning the MEP to a failed state and raising the correct defect condition without delay.

When the **ccm-hold-timer down delay-down** option is configured, the following calculation is used to determine the remote peer time out:  $3.5 \times \text{ccm-interval} + \text{ccm-hold-timer down delay-down}$ .

This command is configured under the association. Only sub-second CCM-enabled MEPs support this hold timer. Ethernet tunnel paths use a similar but slightly different approach and continue to use the existing method. Ethernet tunnels are blocked from using this new hold timer.

It is possible to change this command on the fly without deleting it first. Entering the command with the new values change the values without having to first delete the command.

It is possible to change the **ccm-interval** of a MEP on the fly without first deleting it. This means it is possible to change a sub-second CCM-enabled MEP to 1 second or more. The operator is prevented from changing an association from a sub second CCM interval to a non-sub second CCM interval when a **ccm-hold-timer** is configured in that association. The **ccm-hold-timer** must be removed using the **no** option before allowing the transition from sub second to non-sub second CCM interval.

### 3.4.9 ITU-T Y.1731 ETH-AIS

Alarm Indication Signal (AIS) provides a MEP the ability to signal a fault condition in the reverse direction of the MEP, out the passive side. When a fault condition is detected the MEP generates AIS packets at the configured client levels and at the specified AIS interval until the condition is cleared. Currently a MEP that is configured to generate AIS must do so at a level higher than its own. The MEP configured on the service receiving the AIS packets is required to have the active side facing the receipt of the AIS packet and must be at the same level as the AIS. The absence of an AIS packet for 3.5 times the AIS interval set by the sending a node clear the condition on the receiving MEP.

AIS generation is not subject to the CCM low-priority-defect parameter setting. When enabled, AIS is generated if the MEP enters any defect condition, by default this includes CCM RDI condition.

To prevent the generation of AIS for the CCM RDI condition, the AIS version of the low-priority-defect parameter (under the **ais-enable** command) can be configured to ignore RDI by setting the parameter value to `macRemErrXcon`. The low-priority-defect parameter is specific and influences the protocol under which it is configured. When the low-priority-defect parameter is configured under CCM, it only influences CCM and not AIS. When the low-priority-defect parameter is configured under AIS, it only influences AIS and not CCM. Each protocol can make use of this parameter using different values.

AIS configuration has two components: receive and transmit. AIS reception is enabled when the command **ais-enable** is configured under the MEP. The transmit function is enabled when the **client-meg-level** is configured.

Alarm Indication Signal function is used to suppress alarms at the client (sub) layer following detection of defect conditions at the server (sub) layer. Because of independent restoration capabilities provided within the Spanning Tree Protocol (STP) environments, ETH-AIS is not expected to be applied in the STP environment.

Transmission of frames with ETH-AIS information can be enabled or disabled on a MEP. Frames with ETH-AIS information can be issued at the client MEG Level by a MEP, including a Server MEP, upon detecting the following conditions:

- signal failure conditions in the case that ETH-CC is enabled
- AIS condition in the case that ETH-CC is disabled

For a point-to-point ETH connection at the client (sub) layer, a client layer MEP can determine that the server (sub) layer entity providing connectivity to its peer MEP has encountered defect condition upon receiving a frame with ETH-AIS information. Alarm suppression is straightforward because a MEP is expected to suppress defect conditions associated only with its peer MEP.

For multipoint ETH connectivity at the client (sub) layer, a client (sub) layer MEP cannot determine the specific server (sub) layer entity that has encountered defect conditions upon receiving a frame with ETH-AIS information. More importantly, it cannot determine the associated subset of its peer MEPs for which it should suppress alarms because the received ETH-AIS information does not contain that information.

Therefore, upon receiving a frame with ETH-AIS information, the MEP suppresses alarms for all peer MEPs whether there is still connectivity or not.

Only a MEP, including a server MEP, is configured to issue frames with ETH-AIS information. Upon detecting a defect condition the MEP can immediately start transmitting periodic frames with ETH-AIS information at a configured client MEG Level. A MEP continues to transmit periodic frames with ETH-AIS information until the defect condition is removed. Upon receiving a frame with ETH-AIS information from its server (sub) layer, a client (sub) layer MEP detects AIS condition and suppresses alarms associated with all its peer MEPs. A MEP resumes alarm generation upon detecting defect conditions when AIS condition is cleared.

AIS may also be triggered or cleared based on the state of the entity over which it has been enabled. Including the optional command **interface-support-enable** under the **ais-enable** command tracks the state of the entity and invoke the appropriate AIS action. This means that operators are not required to enable CCM on a MEP to generate AIS if the only requirement is to track the local entity. If a CCM enabled MEP is enabled in addition to this function then both are used to act upon the AIS function. When both CCM and interface support are enabled, a fault in either triggers AIS. To clear the AIS state, the entity must be in an UP operational state and there must be no defects associated with the MEP. The interface support function is available on both service MEPs and facility MEPs both in the Down direction only, with the following exception. An Ethernet QinQ Tunnel Facility MEP does not support interface-support-enable. Many operational models for Ethernet QinQ Tunnel Facility MEPs are deployed with the SAP in the shutdown state.

The following specific configuration information is used by a MEP to support ETH-AIS:

<b>client MEG level</b>	MEG level at which the most immediate client layer MIPs and MEPs exist
<b>ETH-AIS transmission period</b>	determines the transmission period of frames with ETH-AIS information
<b>priority</b>	identifies the priority of frames with ETH-AIS information
<b>drop eligibility</b>	frames with ETH-AIS information are always marked as drop ineligible
<b>interface-support-enable</b>	optional configuration to track the state of the entity over which the MEP is configured
<b>low-priority-defect</b>	optional configuration to exclude the CCM RDI condition from triggering the generation of AIS

A MIP is transparent to frames with ETH-AIS information and therefore does not require any information to support ETH-AIS functionality.

It is important to note that Facility MEPs do not support the generation of AIS to an explicitly configured endpoint. An explicitly configured endpoint is an object that contains multiple individual endpoints, as in pseudowire redundancy.

AIS is enabled under the service and has two parts, receive and transmit. Both components have their own configuration option. The **ais-enable** command under the SAP allows for the processing of received AIS packets at the MEP level. The **client-meg-level** command is the transmit portion that generates AIS if the MEP enter a fault state.

When MEP 101 enters a defect state, it starts to generate AIS out the passive side of the MEP, away from the fault. In this case, the AIS generates out sap 1/1/10:100.31 because MEP 101 is an up MEP on that SAP. The **Defect Flag** indicates that an RDI error state has been encountered. The **Eth-Ais Tx Counted** value is increasing, indicating that AIS is actively being sent.

A single network event may, in turn, cause the number of AIS transmissions to exceed the AIS transmit rate of the network element. A pacing mechanism is in place to assist the network element to gracefully

handle this overload condition. Should an event occur that causes the AIS transmit requirements to exceed the AIS transmit resources, a credit system is used to grant access to the resources. Once all the credits have been used, any remaining MEPs attempting to allocate a transmit resource are placed on a wait list, unable to transmit AIS. If a credit is released, when the condition that caused the MEP to transmit AIS is cleared, a MEP on the wait list consumes the newly available credit. If it is critical that AIS transmit resources be available for every potential event, consideration must be given to the worst case scenario and the configuration should never exceed the potential. Access to the resources and the wait list are ordered and maintained in first come first serve basis.

A MEP that is on the wait list only increments the "Eth-Ais Tx Fail" counter and not the "Eth-Ais TxCount" for every failed attempt while the MEP is on the wait list.

There is no synchronization of AIS transmission state between peer nodes. This is particularly important when AIS is used to propagate fault in ETH-CFM MC-LAG linked designs.

### 3.4.10 ITU-T Y.1731 ETH-CSF

Client signal fail (CSF) is a method that allows for the propagation of a fault condition to a MEP peer, without requiring ETH-CC or ETH-AIS. The message is sent when a MEP detects an issue with the entity in the direction the MEP to its peer MEP. A typical deployment model is an UP MEP configured on the entity that is not executing ETH-CC with its peer. When the entity over which the MEP is configured fails, the MEP can send the ETH-CSF fault message.

To process the reception of the ETH-CSF message, the **csf-enable** function must be enabled under the MEP. When processing of the received CSF message is enabled, the CSF is used as another method to trigger fault propagation, assuming fault propagation is enabled. If CSF is enabled but fault propagation is not enabled, the MEP shows the state of CSF being received from the peer. And lastly, when there is no fault condition, the CSF Rx State displays DCI (Client defect clear) indicating there are no existing failures, even if no CSF has been received. The CSF Rx State indicates the various fault and clear conditions received from the peer during the event.

CSF carries the type of defect that has been detected by the local MEP generating the CSF message.

- 000 – LOS – Client Loss of Signal
- 001 – FDI/AIS – Client forward defect indication
- 010 – RDI – Client reverse defect indication

Clearing the CSF state can be either implicit, time out, or explicit, requiring the client to send the PDU with the clear indicator (011 – DCI – Client defect clear indication). The receiving node uses the multiplier option to determine how to clear the CSF condition. When the multiplier is configured as non-zero (in increments of half seconds between 2 and 30) the CSF is cleared when CSF PDUs have not been received for that duration. A multiplier value of 0 means that the peer that has generated the CSF must send the 011 – DCI flags. There is no timeout condition.

Service-based MEP supports the reception of the ETH-CSF as an additional trigger for the fault propagation process. Primary VLAN and Virtual MEPs do not support the processing of the CSF PDU. CSF is transparent to MIPs. There is no support for the transmission of ETH-CSF packets on any MEP.

### 3.4.11 ITU-T Y.1731 ETH-TST

Ethernet test provides a MEP with the ability to send an in-service on-demand function to test connectivity between two MEPs. The test is generated on the local MEP and the results are verified on the destination

MEP. Any ETH-TST packet generated that exceeds the MTU is silently dropped by the lower level processing of the node.

Specific configuration information required by a MEP to support ETH-test is the following:

- MEG level (MEG level at which the MEP exists)
- unicast MAC address of the peer MEP for which ETH-test is intended
- data (optional element whose length and contents are configurable at the MEP)
- priority (identifies the priority of frames with ETH-Test information)
- drop eligibility (identifies the eligibility of frames with ETHTest information to be dropped when congestion conditions are encountered)

A MIP is transparent to the frames with ETH-Test information and does not require any configuration information to support ETH-Test functionality.

Both nodes require the eth-test function to be enabled to successfully execute the test. Because this is a dual-ended test, initiate on sender with results calculated on the receiver, both nodes need to be checked to see the results.

### 3.4.12 ITU-T Y.1731 ETH-1DM

One-way delay measurement provides a MEP with the ability to check unidirectional delay between MEPs. An ETH-1DM packet is timestamped by the generating MEP and sent to the remote node. The remote node timestamps the packet on receipt and generates the results. The results, available from the receiving MEP, indicate the delay and jitter. Jitter, or delay variation, is the difference in delay between tests. This means the delay variation on the first test is not valid. It is important to ensure that the clocks are synchronized on both nodes to ensure the results are accurate. NTP can be used to achieve a level of clock synchronization between the nodes.



**Note:** Accuracy relies on the nodes ability to timestamp the packet in hardware, and the support of PTP for clock sync.

### 3.4.13 ITU-T Y.1731 ETH-DMM

Two-way delay measurement is similar to one-way delay measurement except it measures the round trip delay from the generating MEP. In this case, clock synchronization issues do not influence the round-trip test results because four timestamps are used. This allows the time it takes for the remote node to process the frame to be removed from the calculation, and as a result, clock variances are not included in the results. The same consideration for first test and hardware based time stamping stated for one-way delay measurement are applicable to two-way delay measurement.

Delay can be measured using one-way and two-way on demand functions. The two-way test results are available single-ended, test initiated, calculation and results viewed on the same node. There is no specific configuration under the MEP on the SAP to enable this function. An example of an on demand test and results are below. The latest test result is stored for viewing. Further tests overwrite the previous results. Delay variation is only valid if more than one test has been executed.



### 3.4.14 ITU-T Y.1731 ETH-SLM



**Note:** Release 9.0 R1 uses pre-standard OpCodes and does not interoperate with any other release or future release.

This synthetic loss measurement approach is a single-ended feature that allows the operator to run on-demand and proactive tests to determine “in”, “out” loss and “unacknowledged” packets. This approach can be used between peer MEPs in both point to point and multipoint services. Only remote MEP peers within the association and matching the unicast destination respond to the SLM packet.

The specification uses various sequence numbers to determine in which direction the loss occurred. Nokia has implemented the required counters to determine loss in each direction. To properly use the information that is gathered the following terms are defined:

- **count**

This is the number of probes that are sent when the last frame is not lost. When the last frames are lost, the count + unacknowledged equals the number of probes sent.

- **out-loss (far-end)**

This represents packets lost on the way to the remote node, from test initiator to test destination.

- **in-loss (near-end)**

This represents packets lost on the way back from the remote node to the test initiator.

- **unacknowledged**

This is the number of packets at the end of the test that were not responded to.

The per probe specific loss indicators are available when looking at the on-demand test runs, or the individual probe information stored in the MIB. When tests are scheduled by Service Assurance Application (SAA) the per probe data is summarized and per probe information is not maintained. Any “unacknowledged” packets are recorded as “in-loss” when summarized.

The on-demand function can be executed from CLI or SNMP. The on demand tests are meant to provide the carrier a means to perform on the spot testing. However, this approach is not meant as a method for storing archived data for later processing. The probe count for on demand SLM has a range of one to 100 with configurable probe spacing between one second and ten seconds. This means it is possible that a single test run can be up to 1000 seconds in length. Although possible, it is more likely the majority of on demand case are run up to 100 probes or less at a one second interval. A node may only initiate and maintain a single active on demand SLM test at any one time. A maximum of one storage entry per remote MEP is maintained in the results table. Subsequent runs to the same peer overwrite the results for that peer. This means when using on demand testing the test should be run and the results checked before starting another test.

The proactive measurement functions are linked to SAA. This backend provides the scheduling, storage and summarization capabilities. Scheduling may be either continuous or periodic. It also allows for the interpretation and representation of data that may enhance the specification. As an example, an optional TLV has been included to allow for the measurement of both loss and delay/jitter with a single test. The implementation does not cause any interoperability because the optional TLV is ignored by equipment that does not support this. In mixed vendor environments loss measurement continues to be tracked but delay and jitter only reports round trip times. It is important to point out that the round trip times in this mixed vendor environment include the remote nodes processing time because only two time stamps are included in the packet. In an environment where both nodes support the optional TLV to include time stamps unidirectional and round trip times are reported. Because all four time stamps are included in the packet, the round trip time in this case does not include remote node processing time. Of course, those

operators that want to run delay measurement and loss measurement at different frequencies are free to run both ETH-SL and ETH-DM functions. ETH-SL is not replacing ETH-DM. Service Assurance is only briefly discussed here to provide some background on the basic functionality.

The ETH-SL packet format contains a test-id that is internally generated and not configurable. The test-id is visible for the on demand test in the display summary. It is possible a remote node processing the SLM frames receive overlapping test-ids as a result of multiple MEPs measuring loss between the same remote MEP. For this reason, the uniqueness of the test is based on remote MEP-ID, test-id and source MAC of the packet.

ETH-SL is applicable to up and down MEPs and as per the recommendation transparent to MIPs. There is no coordination between various fault conditions that could impact loss measurement. This is also true for conditions where MEPs are placed in shutdown state as a result of linkage to a redundancy scheme like MC-LAG. Loss measurement is based on the ETH-SL and not coordinated across different functional aspects on the network element. ETH-SL is supported on service based MEPs.

It is possible that two MEPs may be configured with the same MAC on different remote nodes. This causes various issues in the FDB for multipoint services and is considered a misconfiguration for most services. It is possible to have a valid configuration where multiple MEPs on the same remote node have the same MAC. In fact, this is somewhat likely. Only the first responder is used to measure packet loss. The second responder is dropped, because the same MAC for multiple MEPs is only truly valid on the same remote node.

There is no way for the responding node to understand when a test is completed. For this reason a configurable **inactivity-timer** determines the length of time a test is valid. The timer maintains an active test as long as it is receiving packets for that specific test, defined by the test-id, remote MEP ID and source MAC. When there is a gap between the packets that exceeds the inactivity timer value, the responding node releases the index in the table and responds with a sequence number of 1, regardless of the sequence number sent by the instantiating node. Expiration of this timer causes the reflecting peer to expire the previous test. Packets that follow the expiration of a text are viewed as a new test. The default for the inactivity-timer is 100 second and has a range of ten to 100 seconds.

Only the configuration is supported by HA. There is no synchronization of data between active and standby. Any unwritten, or active tests are lost during a switchover and the data is not recoverable.

ETH-SL provides a mechanism for operators to pro-actively trend packet loss.

### 3.4.15 ITU-T Y.1731 ETH-LMM

The Ethernet Frame Loss Measurement (ETH-LMM) allows the collection of frame counters to determine the unidirectional frame loss between point-to-point ETH-CFM MEP peers. This measurement does not count its own PDU to determine frame loss. The ETH-LMM protocol PDU includes four counters which represent the data sent and received in each direction: Transmit Forward (TxFCf), Receive Forward (RxFCf), Transmit Backward (TxFCb) and the Receive Backward (RxFCb).

The ETH-LMM protocol is designed specifically for point-to-point connections. It is impossible for the protocol to accurately report loss if the point-to-point relationship is broken; for example, if a SAP or MPLS binding receives data from multiple peers, as can be the case in VPLS deployments, this protocol would not be reliable indicator of frame loss.

The loss differential between transmit and receive is determined the first time an LMM PDU is sent. Each subsequent PDU for a specific test performs a computation of differential loss from that epoch. Each processing cycle for an LMR PDU determines if there is a new maximum or minimum loss window, adds any new loss to the frame loss ratio computation, and updates the four raw transmit and receive counters. The individual probe results are not maintained; these results are only used to determine a new minimum

or maximum. A running total of all transmit and receive values is used to determine the average Frame Loss Ratio (FLR) at the completion of the measurement interval. The data set includes the protocol information in the opening header, followed by the frame counts in each direction, and finally the FLR percentages.

The user must understand the restrictions of service before selecting this method of loss measurement. Statistics are maintained per forwarding complex. Multiple path environments may spread frames between the same two peers across different forwarding complexes (for example, link aggregation groups). The ETH-LMM protocol has no method to rationalize different transmit and receive statistics when there are complex changes or when any statistics are cleared on either of the peer entities. The protocol resynchronizes but the data collected for that measurement interval is invalid. The protocol has no method to determine if the loss is true loss or whether some type of complex switch has occurred or statistics were cleared. Consequently, the protocol cannot use any suspect flag to mark the data as invalid. Higher level systems must coordinate network events and administrative actions that can cause the counters to become non-representative of the service data loss.

Packet reordering also affect frame loss and gain reporting. If there is queuing contention on the local node or if path differences in the network cause interleaved or delayed frames, the counter stamped into the LMM PDU can introduce frame gain or loss in either direction. For example, if the LMM PDU is stamped with the TxFCf counter and the LMM PDU traffic is interleaved, the interleaving cannot be accounted for in the counter and a potential gain is realized in the forward direction. This is because the original counter included as the TxFCf value does not include the interleaved packets and the RxFCf counter on the remote peer includes them. Gains and losses even out over the life of the measurement interval. Absolute values are used for any negative values, per interval or at the end of the measurement interval.

Launching a single-ended test is under the control of the OAM Performance Monitoring (OAM-PM) architecture, and the test adheres to the rules of OAM-PM. The ETH-LMM functionality is only available under the OAM-PM configuration. This feature is not available through interactive CLI or SAA. OAM-PM requires the configuration of a test ID for all OAM-PM tests. The ETH-LMM protocol does not define the necessity for this ID, nor does it carry the 4-byte test ID in the packet. This is for local significance and uniformity with other protocols under the control of the OAM-PM architecture.

Support is included for point-to-point Up and Down Service MEPs and Down Facility MEPs (port, LAG, and base router interfaces). Base router interface accuracy may be affected by the Layer 2 or Layer 3 interworking functions, routing protocol, ACLs, QoS policies, and other Layer 3 functions that were never meant to be accounted for by an Ethernet frame loss measurement tool. Launch functions require IOM/IMM or later, as well as a SF/CPM3 or later.

Resource contention extends beyond the sharing of common LMM resources used for packet counting and extraction. There is also protocol-level contention. For example, Cflowd cannot be counted or sampled on an entity that is collecting LMM statistics. Collection of statistics per Ethernet SAP, per MPLS SDP binding, or per facility is not enabled by default.

ETH-LMM is not supported in the following models:

- up MEPs in an I-VPLS or PBB Epipe that crosses a PBB infrastructure. This configuration results in LMM PDUs being discarded on the remote BVPLS node.
- ETH-LMM when primary VLANs are configured against the MEP
- nonoperational SAP or MPLS SDP bindings over which the Up MEP is configured. This configuration causes LMM or LMR transmissions to fail because the SAP which stores the counters is unavailable to the LMM PDU.

QinQ tunnel collection is the aggregate of all outer VLANs that share the VLAN with the tunnel. If the QinQ is configured to collect LMM statistics, then any service MEP that shares the same VLAN as the QinQ tunnel is blocked from configuring the respective **collect-imm-stats** command. The reverse is also true; if

a fully qualified SAP is configured to collect LMM statistics, the QinQ tunnel that shares the outer VLAN is blocked from configuring the respective **collect-lmm-stats** command.

QoS models contribute significantly to the accuracy of the LMM counters. If the QoS function is beyond the LMM counting function, it can lead to mismatches in the counter and transmit and receive information.

### 3.4.15.1 ETH-LMM single SAP counter

A single LMM counter per SAP or per MPLS SDP binding or per facility counter is the most common option for deployment of the LMM frame-based counting model. This single counter model requires careful consideration for the counter location. Counter integrity is lost when counting incurs entity conflicts, as is typical in facility MEP and service MEP overlap. The operator must choose one type of facility MEP or the service MEP. If a facility MEP is chosen (Port, LAG, QinQ Tunnel or Base Router Interface) care must be taken to ensure the highest configured MEP performs the loss collection routine.

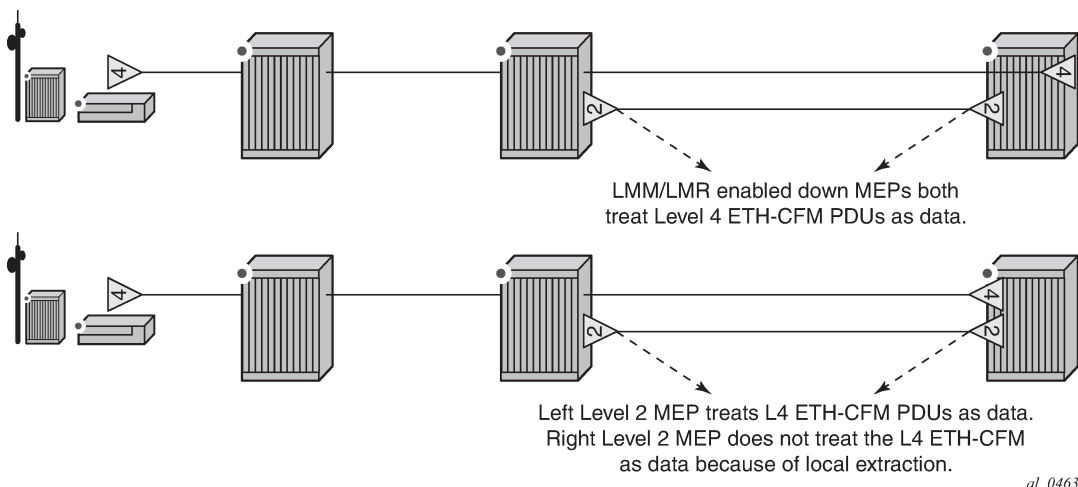
Configuring loss collection on a lower level MEP leads to additive gain introduced in both directions. Although the collection statement is not blocked by CLI or SNMP when there are potential conflicts, only one can produce accurate results. The operator must be aware of lower level resource conflicts. For example, a null based service SAP, any default SAP context or SAP that covers the entire port or facility resource, such as sap 1/1/1, always counts the frame base loss counter against the SAP and never the port, regardless of the presences of a MEP or the **collect-lmm-stats** configuration on the SAP. Resource contention extends beyond the sharing of common resources used for packet counting and extraction.

For this feature to function with accurate measurements, the **collect-lmm-stats** is required under the ETH-CFM context for the Ethernet SAP or MPLS SDP binding or under the MEP in the case of the facility MEP. If this command is not enabled on the launch and reflector, the data in the ETH-LMM and ETH-LMR PDU is not representative and the data captured is invalid.

The **show>service>sdp-using eth-cfm** and **show>service>sap-using eth-cfm** commands have been expanded to include the **collect-lmm-stats** option for service based MEPs. The **show>eth-cfm>cfm-stack-table facility** command has been expanded to include **collect-lmm-stats** to view all facility MEPs. Using these commands with this new option displays the entities that are currently collecting LMM counter.

The counter includes all frames that are transmitted or received regardless of class of service or discard eligibility markings. Locally transmitted and locally terminated ETH-CFM frames on the peer collecting the statistics are not included in the counter. However, there are deployment models that introduce artificial frame loss or gain when the ETH-CFM launch node and the terminating node for some ETH-CFM packets are not the same peers. [Figure 50: Mismatched LMM statistical counters](#) demonstrates this issue.

Figure 50: Mismatched LMM statistical counters



### 3.4.15.2 ETH-LMM per forwarding class counter

Frame loss measurement can be deployed per forwarding class (FC) counter. The **config>oam-pm>session>ethernet>lmm>enable-fc-collection** command in the related oam-pm session enables frames to be counted on an FC basis, either in or out of profile. This counting method alleviates some of the ordering and interleaving issues that arise when using a single counter, but does not improve on the base protocol concerns derived from multiple paths and complex based counting.

This approach requires the operator to configure the individual FCs of interest and the profile status of the frames under the **collect-lmm-fc-stats** context. The command allows for the addition or removal of an individual FC by using a differential. The entire command with the wanted FC statements must be included. The system determines the new, deleted, and unchanged FCs. New FCs are allocated a counter. Deleted FCs stop counting. Unchanged FCs continue counting.

Support for per-FC collection includes SAPs, MPLS SDP bindings, and router interfaces.

The **enable-fc-collection** command must be coordinated between the ETH-LMM test and counting model to configure either single per SAP or MPLS SDP binding counter, or per FC counter. The command is disabled by default, and single per SAP or MPLS SDP binding counter is used.

Symmetrical QoS is required for correct collection of frame counters. The FC must match the priority of the OAM-PM ETH-LMM test. The ETH-LM PDUs must ensure that they are mapped to the correct FC on ingress and egress so that the appropriate counters are collected. Mismatches between the ETH-LMM PDUs and the collected FC cause incorrect or no data to be reported.

The **show>eth-cfm>collect-lmm-fc-stats** command displays the SAPs, MPLS SDP bindings, and router interfaces that are configured for per-FC collection, and whether the collection is priority aware or unaware. It also includes the base mapping of OAM-PM ETH-LMM priority to FC.

### 3.4.15.3 Interaction between single and per FC counters

Entities that support LMM collection may only use one of the following collection models:

- single counter (**collect-lmm-stats**)

- per FC counter (**collect-lmm-fc-stats**)

The **collect-lmm-stats** and **collect-lmm-fc-stats** commands are mutually exclusive.

OAM-PM rejects ETH-LMM test configurations from same source MEPs that have different **enable-fc-collection** configurations.

Ensure that the LMM collection model that is configured on the entity (**collect-lmm-stats** or **collect-lmm-fc-stats**) matches the configuration of the **enable-fc-collection** command within the OAM-PM session, and that the priority of the test maps to the required FC.

### 3.4.16 ETH-CFM destination options

ETH-CFM relies on Ethernet addressing and reachability. ETH-CFM destination addressing may be derived from the Ethernet encapsulation, or may be a target address within the ETH-CFM PDU. Addressing is the key to identifying both the source and the destination management points (MPs).

The SR OS implementation dynamically assigns the MP MAC address using the appropriate pool of available hardware addresses on the network element, which simplifies the configuration and maintenance of the MP. The MP MAC address is tied to the specific hardware element, and its addressing can change when the associated hardware is changed.

The optional **mac-address** *mac-address* configuration command can be used to eliminate the dynamic nature of the MEP MAC addressing. This optional configuration associates a configured MAC address with the MEP in place of dynamic hardware addressing. The optional **mac-address** configuration is not supported for all service types.

ETH-CFM tests can adapt to changing destination MAC addressing by using the **remote-mepid** *mep-id* command in place of the unicast statically-configured MAC address. SR OS maintains a learned remote MAC table (visible by using the **show>eth-cfm>learned-remote-mac** command) for all MEPs that are configured to use ETH-CC messaging. Usually, when the **remote-mepid** *mep-id* command is used as part of a supported test function, the test searches the learned remote MAC table for a unicast address that associates the local MEP and the requested remote MEP ID. If a unicast destination address is found for that relationship, it is used as the unicast destination MAC address.

The learned remote MAC table is updated and maintained by the ETH-CC messaging process. When an address is learned and recorded in the table, it is maintained even if the remote peer times out or the local MEP is shut down. The address is not maintained in the table if the **remote-mepid** statement is removed from the associated context by using the **no remote-mepid** *mep-id* command for a peer. The CCM database clears the peer MAC address and enters an all-0 MAC address for the entry when the peer times out. The learned remote MAC table maintains the previously learned peer MAC address. If an entry must be deleted from the learned remote MAC table, the **clear>learned-remote-mac [mep mep-id [remote-mepid mep-id]] domain md-index association ma-index** command can be used. Deleting a local MEP removes the local MEP and all remote peer relationships, including the addresses previously stored in the learned remote MAC table.

The individual ETH-CFM test scheduling functions that use the **remote-mepid** *mep-id* option have slightly different operational behaviors.

Global interactive CFM tests support the **remote-mepid** *mep-id* option as an alternative to *mac-address*. A test only starts if a learned remote MAC table contains a unicast MAC address for the remote peer, and runs to completion with that MAC address. If the table does not contain the required unicast entry associated with the specified remote MEP ID, the test fails to start.

SAA ETH-CFM test types support the **remote-mepid** *mep-id* option as an alternative to *mac-address*. If, at the scheduled start of the individual run, the learned remote MAC table contains a unicast learned remote

MAC address for the remote peer, the test runs to completion with the initial MAC address. If the table does not contain the required entry, the test terminates after the lesser window of either the full test run or 300 s. A run that cannot successfully determine a unicast MAC address designates the last test result as "failed". If a test is configured with the **continuous** configuration option, it is rescheduled; otherwise, the test is not rescheduled.

OAM-PM Ethernet test families, specifically DMM, SLM, and LMM, support the **remote-mepid** *mep-id* option as an alternative to the **dest-mac** *ieee-address* configuration. If the learned remote MAC table contains a unicast learned remote MAC address for the remote peer, the test uses this MAC address as the destination. OAM-PM adapts to changes for MAC addressing during the measurement interval when the **remote-mepid** *mep-id* option is configured. It should be expected that the measurement interval includes update-induced PM errors during the transition. If the table does not contain the required entry, the test does not attempt to transmit test PDUs, and presents the "Dest Remote MEP Unknown" detectable transmission error.

### 3.4.17 ITU-T Y.1731 ETH-BN

The Ethernet Bandwidth Notification (ETH-BN) function is used by a server MEP to signal changes in link bandwidth to a client MEP.

This functionality is for point-to-point microwave radios to modify the downstream traffic rate toward the microwave radio to match its microwave link rate. When a microwave radio uses adaptive modulation, the capacity of the radio can change based on the condition experienced by the microwave link. For example, in adverse weather conditions that cause link degradation, the radio can change its modulation scheme to a more robust one (which reduces the link bandwidth) to continue transmitting. This change in bandwidth is communicated from the server MEP on the radio, using ETH-BN Message (ETH-BNM), to the client MEP on the connected router. The server MEP transmits periodic messages with ETH-BN information including the interval, the nominal, and currently available bandwidth. A port MEP with the ETH-BN feature enabled processes the information in the CFM PDU. The operational port egress rate can be modified to adjust the rate of traffic sent to the radio.

A port MEP supports the client side reception and processing of the ETH-BNM sent by the server MEP. By default, processing is disabled. A port that can process an ETH-BNM is a configuration specific to that port, even when the port is a LAG member port. The ETH-BN configuration on the LAG member ports does not have to be the same. However, mismatches in the configuration on these member ports could lead to significant differences in operational egress rates within the same LAG. Different operational rates on the LAG member ports as a result of ETH-BN updates are not considered when hashing packets to the LAG member ports.

The **no** form of the **configure port ethernet eth-cfm mep eth-bn receive** CLI command sets the ETH-BN processing state on the port MEP. A port MEP supports untagged packet processing of ETH-CFM PDUs at domain levels 0 and 1 only. The port client MEP sends the ETH-BN rate information received to be applied to the port egress rate in a QoS update. A pacing mechanism limits the number of QoS updates sent. The **configure port ethernet eth-cfm mep eth-bn rx-update pacing** CLI command allows the updates to be paced using a configurable range of 1 to 600 seconds (the default is 5 seconds). The pacing timer begins to countdown following the most recent QoS update sent to the system for processing. When the timer expires, the most recent update that arrived from the server MEP is compared to the most recent value sent for system processing. If the value of the current bandwidth is different from the previously processed value, the update is sent and the process begins again. Updates with a different current bandwidth that arrive when the pacing timer has already expired are not subject to a timer delay. See the *7450 ESS*, *7750 SR*, *7950 XRS*, and *VSR Interface Configuration Guide* for more information about these commands.

A complementary QoS configuration is required to allow the system to process nominal bandwidth updates from the CFM engine. The **configure port ethernet no eth-bn-egress-rate changes** CLI command is required to enable the QoS function to update the port egress rates based on the current available bandwidth updates from the CFM engine. By default, the function is disabled.

Both the CFM and the QoS functions must be enabled for the changes in current bandwidth to dynamically update the egress rate.

When the MEP enters a state that prevents it from receiving the ETH-BNM, the current bandwidth last sent for processing is cleared and the egress rate reverts to the configured rate. Under these conditions, the last update cannot be guaranteed as current. Explicit notification is required to dynamically update the port egress rate. The following types of conditions lead to ambiguity:

- administrative MEP shutdown
- port admin down
- port link down
- **eth-bn no receive** transitioning the ETH-BN function to disable

If the **configure port ethernet eth-bn-egress-rate-changes** command is disabled using the **no** option, CFM continues to send updates, but the updates are held without affecting the port egress rate.

The ports supporting ETH-BN MEPs can be configured for network, access, or hybrid modes. When ETH-BN is enabled on a port MEP and the **config>port>ethernet>eth-cfm>mep>eth-bn>receive** and the QoS **config>port>ethernet>eth-bn-egress-rate-changes** contexts are configured, the egress rate is dynamically changed based on the current available bandwidth indicated by the ETH-BN server.

The port egress rate is capped by the minimum of the configured egress rate and the maximum port rate. The minimum egress rate is one kbyte. If a current bandwidth of zero is received, it does not affect the egress port rate and the previously processed current bandwidth continues to be used.

The client MEP requires explicit notification of changes to update the port egress rate. The system does not timeout any previously-processed current bandwidth rates using a timeout condition. The specification does allow a timeout of the current bandwidth if a message has not been received in 3.5 times the ETH-BN interval. However, the implicit approach can lead to misrepresented conditions and has not been implemented.

When starting or restarting the system, the configured egress rate is used until an ETH-BNM arrives on the port with a new bandwidth request from the ETH-BN server MEP.

An event log is generated each time the egress rate is changed based on reception of an ETH-BNM. If an ETH-BNM is received that does not result in a bandwidth change, no event log is generated.

The destination MAC address can be a Class 1 multicast MAC address (that is, 01-80-C2-00-0x) or the MAC address of the port MEP configured. Standard CFM validation and identification must be successful to process any CFM PDU.

For information about the **eth-bn-egress-rate-changes** command, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Interface Configuration Guide*.

The PDU used for ETH-BN information is called the Bandwidth Notification Message (BNM). It is identified by a sub-OpCode within the Ethernet Generic Notification Message (ETH-GNM).

[Table 14: BNM PDU format fields](#) shows the BNM PDU format fields.



Table 14: BNM PDU format fields

Label	Description
MEG Level	Carries the MEG level of the client MEP (0 to 7). This field must be set to either 0 or 1 to be recognized as a port MEP.
Version	The current version is 0.
OpCode	The value for this PDU type is GNM (32).
Flags	Contains one information element: Period (3 bits) to indicate how often ETH-BNM messages are transmitted by the server MEP. Valid values are: <ul style="list-style-type: none"> <li>• 100 (1 frame/s)</li> <li>• 101 (1 frame/10 s)</li> <li>• 110 (1 frame/min)</li> </ul>
TLV Offset	This value is set to 13.
Sub-OpCode	The value for this PDU type is BNM (1).
Nominal Bandwidth	The nominal full bandwidth of the link, in Mbytes/s. This information is reported in the display but not used to influence QoS egress rates.
Current Bandwidth	The current bandwidth of the link in Mbytes/s. The value is used to influence the egress rate.
Port ID	A non-zero unique identifier for the port associated with the ETH-BN information, or zero if not used. This information is reported in the display, but is not used to influence QoS egress rates.
End TLV	An all zeros octet value.

The **show eth-cfm mep eth-bandwidth-notification** display output includes the ETH-BN values received and extracted from the PDU, including the last reported value and the pacing timer. If n/a appears in the field, it means that field has not been processed.

The base **show eth-cfm mep** output is expanded to include the disposition of the ETH-BN receive function and the configured pacing timer.

The **show port port-id detail** is expanded to include an ETH-BNM section. This section includes the egress rate disposition and the current egress BN rate being used.

### 3.4.18 ETH-CFM statistics

A number of statistics are available to view the current overall processing requirements for CFM. Any packet that is counted against the CFM resource is included in the statistics counters. These counters

do not include the counting of sub-second CCM, ETH-CFM PDUs that are generated by non-ETH-CFM functions (which includes OAM-PM and SAA) or are filtered by an applicable security configuration.

SAA and OAM-PM use standard CFM PDUs. The reception of these packets are included in the receive statistics. However, these two functions are responsible for launching their own test packets and do not consume ETH-CFM transmission resources.

Per system and per MEP statistics are available with a per OpCode breakdown. Use the **show>eth-cfm>statistics** command to view the statistics at the system level. Use the **show>eth-cfm>mep mep-id domain md-index association ma-index statistics** command to view the per MEP statistics. These statistics may be cleared by substituting the **clear** command for the **show** command. The clear function only clears the statistics for that function. For example, clearing the system statistics does not clear the individual MEP statistics, each maintain their own unique counters.

All known OpCodes are listed in transmit and receive columns. Different versions for the same OpCode are not distinguished for this display. This does not imply the network element supports all listed functions in the table. Unknown OpCodes are dropped.

It is also possible to view the top ten active MEPs on the system. The term active can be defined as any MEP that is in a **no shutdown** state. The **tools dump eth-cfm top-active-meeps** command can be used to see the top ten active MEPs on the system. The counts are based from the last time to command was issued with the **clear** option. MEPs that are in a shutdown state are still terminating packets, but these do not appear on the active list.

These statistics help operators to determine the busiest active MEPs on the system as well a breakdown of per OpCode processing at the system and MEP level.

### 3.4.19 ETH-CFM packet debug

The debug infrastructure supports the decoding of both received and transmitted valid ETH-CFM packets for MEPs and MIPs that have been tagged for decoding. The **eth-cfm** hierarchy has been added to the existing **debug** CLI command tree. When a MEP or MIP is tagged by the debug process, valid ETH-CFM PDUs are decoded and presented to the logging infrastructure for operator analysis. Fixed queue limits restrict the overall packet rate for decoding. The receive and transmit ETH-CFM debug queues are serviced independently. Receive and transmit correlation is not guaranteed across the receive and transmit debug queues. The **tools dump eth-cfm debug-packet** command displays message queue exceptions.

Valid ETH-CFM packets must pass a multiple-phase validity check before being passed to the debug parsing function. The MAC addresses must be non-zero. If the destination MAC address is multicast, the last nibble of the multicast address must match the expected level of a MEP or MIP tagged for decoding. Packet length and TLV formation, usage, and, where applicable, field validation are performed. Finally, the OpCode-specific TLV structural checks are performed against the remainder of the PDU.

An ETH-CFM packet that passes the validation process is passed to the debug decoding process for tagged MEPs or MIPs. The decoding process parses the PDU for analysis. Truncation of individual TLVs occurs when:

- TLV processing requires multiple functions; this occurs with TLVs that include sub-fields
- an Organizational Specific TLV exists
- padding has been added, as in the case of the optional Data or Test TLVs
- an unknown OpCode is detected; the decode procedure processes the generic ETH-CFM header with a hex dump for unknown fields and TLVs

The number of printable bytes is dependent on the reason for truncation.

Any standard fields in the PDU that are defined for a specific length with a Must Be Zero (MBZ) attribute in the specification are decoded based on the specification field length. There is no assumption that packets adhere to the MBZ requirement in the byte field; for example, the MEP-ID is a 2-byte field with three reserved MBZ bits, which translates into a standard MEP-ID range of 0 to 8191. If the MBZ bits are violated, then the 2-byte field is decoded using all non-zero bits in the 2-byte field.

The decoding function is logically positioned between ETH-CFM and the forwarding plane. Any ETH-CFM PDU discarded by an applicable security configuration is not passed to the debug function. Any packet that is discarded by squelching (using the **config>service>sap>eth-cfm>squelch-ingress-levels** and **config>service>sap>eth-cfm>squelch-ingress-ctag-levels** commands) or CPU protection (using the **config>service>sap>eth-cfm>cpu-protection eth-cfm-monitoring** command), bypasses the decoding function. Care must be taken when interpreting specific ETH-CFM PDU decodes. Those PDUs that have additional, subsequent, or augmented information applied by the forwarding mechanisms may not be part of the decoded packet. Augmentation includes the timestamp (the stamping of hardware based counters [LMM]) applied to ETH-CFM PDUs by the forwarding plane.

This function allows for enhanced troubleshooting for ETH-CFM PDUs to and from tagged MEPs and MIPs. Only defined and node-supported functionality are decoded, possibly with truncation. Unsupported or unknown functionality on the node is treated on a best-effort basis, typically handled with a decode producing a truncated number of hex bytes.

This functionality does not support decoding of sub-second CCM, or any ETH-CFM PDUs that are processed by non-ETH-CFM entities (which includes SAA CFM transmit functions), or MIPs created using the **default-domain** table.

### 3.4.20 ETH-CFM CoS considerations

UP MEPs and Down MEPs have been aligned to better emulate service data. When an UP MEP or DOWN MEP is the source of the ETH-CFM PDU, the priority value configured, as part of the configuration of the MEP or specific test, is treated as the Forwarding Class (FC) by the egress QoS policy. The numerical ETH-CFM **priority** value resolves FCs using the following mapping:

- 0 — be
- 1 — l2
- 2 — af
- 3 — l1
- 4 — h2
- 5 — ef
- 6 — h1
- 7 — nc

If there is no egress QoS policy, the priority value is mapped to the CoS values in the frame. An ETH-CFM frame utilizing VLAN tags has the DEI bit mark the frame as "discard ineligible". However, egress QoS Policy may overwrite this original value. The Service Assurance Agent (SAA) uses **[fc {fc-name} [profile {in | out}]]** to accomplish similar functionality.

UP MEPs and DOWN MEPs terminating an ETH-CFM PDU use the received FC as the return priority for the appropriate response, again feeding into the egress QoS policy as the FC.

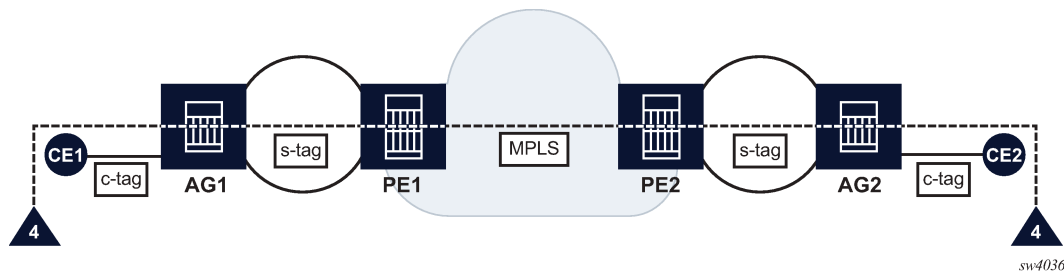
This does not include Ethernet Linktrace Response (ETH-LTR). The specification requires the highest priority on the bridge port should be used in response to an Ethernet Linktrace Message (ETH-LTM). This

provides the highest possible chance of the response returning to the source. Operators may configure the linktrace response priority of the MEP using the **ccm-ltm-priority**. MIPs inherit the MEPs priority unless the **mip-ltr-priority** is configured under the bridging instance for the association (**config>eth-cfm>domain>assoc>bridge**).

### 3.4.21 Silent CFM dropping with squelching

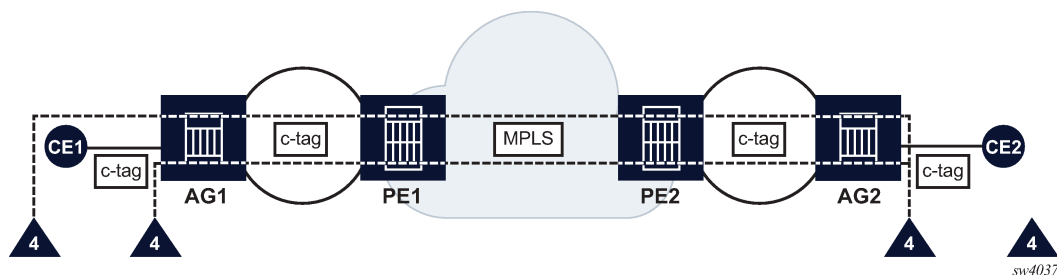
The ETH-CFM architecture defines the hierarchy that supports separation of Ethernet CFM OAM domains of responsibility. Typically, encapsulation methods are used to tunnel traffic transparently through intermediate segments. Using a network topology as shown in [Figure 51: ETH-CFM CPE to CPE](#), CE traffic arrives at the aggregation node and is encapsulated with a service-provider tag which hides the customer-specific tag as the packet moves through segments of the network. This method treats the ETH-CFM traffic in the same manner. The application of additional tags prevents ETH-CFM conflicts in the various Ethernet CFM OAM domains, even if the domain levels, in this case, four in this example are reused.

Figure 51: ETH-CFM CPE to CPE



In some scenarios, this additional tagging principle is not displayed and this may result in conflicts and collisions. For example, as shown in [Figure 52: ETH-CFM collision between OAM domains](#), an additional pair of Domain Level 4 UP MEPs are configured on the aggregation nodes. These aggregation nodes are c-tag aware. The ETH-CFM packets that are transmitted from the CE pass transparently through the passive side of the MEP (the side facing away from the ETH-CFM packet transmission) and arrive on the active side of the unintended peer MEP. This could cause a number of defect conditions to occur on the unexpectedly terminating MEP and on the unreachable MEP. For simplicity, only the direction from left CE to right side of the network is shown, although the problem exists in both directions.

Figure 52: ETH-CFM collision between OAM domains

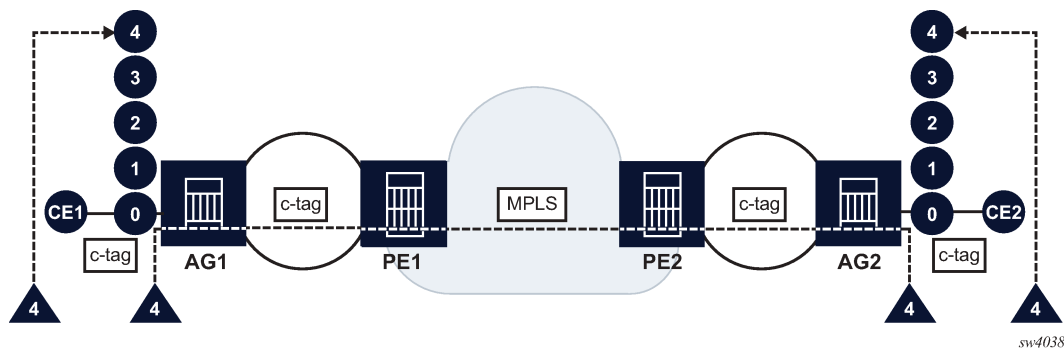


These issues can be resolved through communication of ETH-CFM domain-level ownership using a business agreement. However, this communication is only a business agreement and could be violated by

misconfiguration. Network-level enforcement of this agreement is important to protect both the ETH-CFM OAM domains of responsibility.

ETH-CFM ingress squelching capabilities are available to enforce the agreement and prevent unwanted ETH-CFM packets from entering a domain of responsibility that should not be exposed to ETH-CFM packets from outside its domain. [Figure 53: Enforcement of the CFM level agreement using squelching](#) shows the generic enforcement of the agreement using squelching. In this agreement, Domain Levels 4 and lower are reserved by the provider of the EVC. Domain Levels 5 and above are outside the EVC provider's scope and must pass transparently through the Ethernet CFM OAM domain. The EVC provider's boundaries are configured to enforce this agreement and silently discard all ETH-CFM packets that arrive on the ingress points of the boundary at Domain Level 4 and below.

*Figure 53: Enforcement of the CFM level agreement using squelching*



Two different squelch functions are supported, using the **squelch-ingress-levels** command and the **squelch-ingress-ctag-levels** command.

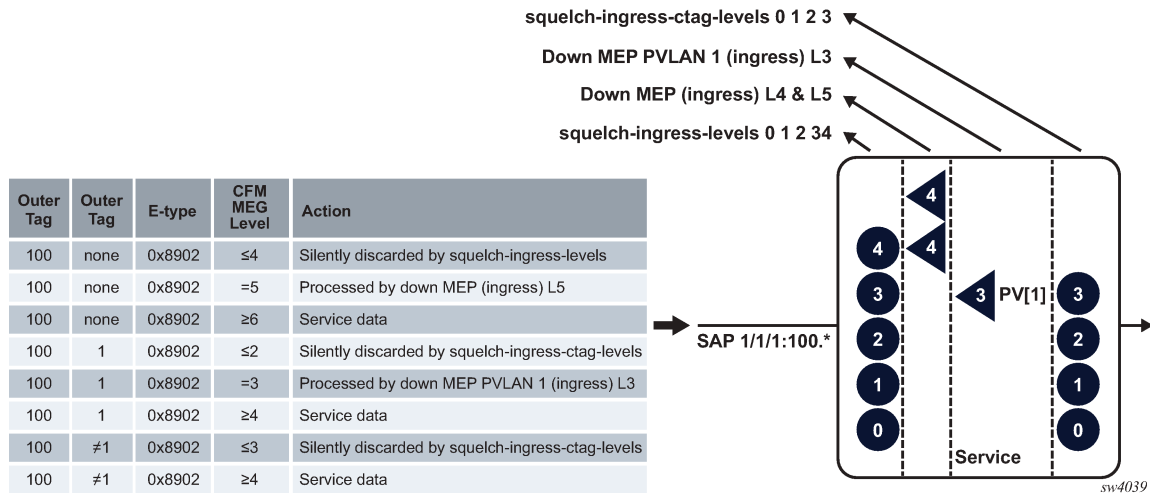
The **squelch-ingress-levels** command configures an exact service delineation SAP and binding match at the ingress followed immediately by Ethernet type 0x8902. This configuration silently discards the appropriate ETH-CFM packets according to the configured levels of the command, regardless of the presence of an ingress ETH-CFM management point, MEP or MIP. This squelch function occurs before other ETH-CFM packet processing functions.

The **squelch-ingress-ctag-levels** command is supported for Epipe and VPLS services only. It configures an exact service delineation SAP and binding match of the ingress skipping one addition tag at the ingress, for a maximum total tag length of two tags, followed by Ethernet type 0x8902. This configuration silently discards the appropriate ETH-CFM packets according to the levels that match the configured squelch levels, if an addition tag beyond the service delineation exists. It ignores the value of the additional tag exposing that entire range to this squelch function, if there is no ingress ETH-CFM management point, MEP or MIP, at one of the configured levels covered by the squelch configuration. This squelch function is different from the option configured by **squelch-ingress-levels**, because it occurs after the processing of an ingress MEP or ingress MIP configured with a primary VLAN within the configured squelch levels. When a primary VLAN ingress MEP or ingress MIP is configured at a VLAN within the squelch level, that entire primary VLAN ETH-CFM function follows regular ETH-CFM primary VLAN rules. In this configuration, the ingress ETH-CFM packets that do not have an ingress MEP or ingress MIP configured for that VLAN are exposed to the squelching rules instead of the primary VLAN rules of ETH-CFM processing. In this case, ETH-CFM primary VLAN ingress processing occurs before the **squelch-ingress-ctag-levels** functions.

Both variants can be configured together on supported connections and within their supported services. [Figure 54: Logical processing chain and interaction](#) shows the logical processing chain and interaction using an ingress QinQ SAP in the form VID.\* and various ingress ETH-CFM MEPs. Although not shown

in the [Figure 54: Logical processing chain and interaction](#), the processing rules are the same for ingress MIPs, which are ETH-LBM (loopback) and ETH-CFM-LTM (linktrace) aware.

Figure 54: Logical processing chain and interaction



There is no requirement to configure ingress MEPs or ingress MIPs if the goal is simply to silently discard ETH-CFM packets matching a domain level criterion. The squelch commands require contiguous levels configuration.

## 3.5 OAM mapping

OAM mapping is a mechanism that enables a way of deploying OAM end-to-end in a network where different OAM tools are used in different segments. For instance, an Epipe service could span across the network using Ethernet access (CFM used for OAM), pseudowire (T-LDP status signaling used for OAM), and Ethernet access (E-LMI used for OAM).

In the SR OS implementation, the Service Manager (SMGR) is used as the central point of OAM mapping. It receives and processes the events from different OAM components, then decides the actions to take, including triggering OAM events to remote peers.

Fault propagation for CFM is by default disabled at the MEP level to maintain backward compatibility. When required, it can be explicitly enabled by configuration.

Fault propagation for a MEP can only be enabled when the MA comprises no more than two MEPs (point-to-point).

Fault propagation cannot be enabled for eth-tun control MEPs (MEPs configured under the eth-tun primary and protection paths). However, failure of the eth-tun (meaning both paths fail) is propagated by SMGR because all the SAPs on the eth-tun go down.

### 3.5.1 CFM connectivity fault conditions

CFM MEP declares a connectivity fault when its defect flag is equal to or higher than its configured lowest defect priority. The defect can be any of the following depending on configuration.

- **DefRDICCM**

This is Remote Defect Indication. Remote MEP is declaring a fault by setting the RDI bit in the CCM PDU. Typically a result of raising a local defect based on of the CCM or lack of CCM from an expected or unexpected peer. A feedback loop into the association as a notification because CCM is multicast message with no response.

- **DefMACstatus**

This indicates a MAC layer issue. Remote MEP is indicating remote port or interface status not operational.

- **DefRemoteCCM**

This indicates there is no communication from remote peer. MEP not receiving CCM from an expected remote peer. Timeout of CCM occurs in 3.5 x CC interval.

- **DefErrorCCM**

This indicates remote configuration does not match local expectations. Receiving CC from remote MEP with inconsistent timers, lower MD/MEG level within same MA/MEG, MEP receiving CCM with its own MEP ID within same MA/MEG.

- **DefXconCCM**

Cross-connected services. MEP receiving CCM from different MA/MEG.

- **Reception of AIS for the local MEP level**

This is an additional fault condition that also applies to Y.1731 MEPs.

Setting the lowest defect priority to allDef may cause problems when fault propagation is enabled in the MEP. In this scenario, when MEP A sends CCM to MEP B with interface status down, MEP B responds with a CCM with RDI set. If MEP A is configured to accept RDI as a fault, then it gets into a dead lock state, where both MEPs declare fault and are never be able to recover. The default lowest defect priority is DefMACstatus. In general terms, when a MEP propagates fault to a peer the peer receiving the fault must not reciprocate with a fault back to the originating MEP with a fault condition equal to or higher than the originating MEP low-priority-defect setting. It is also very important that different Ethernet OAM strategies should not overlap the span of each other. In some cases, independent functions attempting to perform their normal fault handling can negatively impact the other. This interaction can lead to fault propagation in the direction toward the original fault, a false positive, or worse, a deadlock condition that may require the operator to modify the configuration to escape the condition. For example, overlapping Link Loss Forwarding (LLF) and ETH-CFM fault propagation could cause these issues.

### 3.5.2 CFM fault propagation methods

When CFM is the OAM module at the other end, it is required to use any of the following methods (depending on local configuration) to notify the remote peer:

- generating AIS for specific MEP levels
- sending CCM with interface status TLV "down"
- stopping CCM transmission

For using AIS for fault propagation, AIS must be enabled for the MEP. The AIS configuration needs to be updated to support the MD level of the MEP (currently it only supports the levels above the local MD level).

Note that the existing AIS procedure still applies even when fault propagation is disabled for the service or the MEP. For example, when a MEP loses connectivity to a configured remote MEP, it generates AIS if

it is enabled. The new procedure that is defined in this document introduces a new fault condition for AIS generation, fault propagated from SMGR, that is used when fault propagation is enabled for the service and the MEP.

The transmission of CCM with interface status TLV is triggered and does not wait for the expiration of the remaining CCM interval transmission. This rule applies to CFM fault notification for all services.

For a specific SAP and SDP-binding, CFM and SMGR can only propagate one single fault to each other for each direction (up or down).

When there are multiple MEPs (at different levels) on a single SAP and SDP-binding, the fault reported from CFM to SMGR is the logical OR of results from all MEPs. Basically, the first fault from any MEP is reported, and the fault is not cleared as long as there is a fault in any local MEP on the SAP and SDP-binding.

### 3.5.3 Epipe services

Down and up MEPs are supported for Epipe services as well as fault propagation. When there are both up and down MEPs configured in the same SAP and SDP-binding and both MEPs have fault propagation enabled, a fault detected by one of them is propagated to the other, which in turn propagates fault in its own direction.

### 3.5.4 CFM detected fault

When a MEP detects a fault and fault propagation is enabled for the MEP, CFM needs to communicate the fault to SMGR, so SMGR marks the SAP or SDP-binding faulty but still oper up. CFM traffic can still be transmitted to or received from the SAP and SDP-binding to ensure when the fault is cleared, the SAP goes back to normal operational state. Because the operational status of the SAP and SDP-binding is not affected by the fault, no fault handling is performed. For example, applications relying on the operational status are not affected.

If the MEP is an up MEP, the fault is propagated to the OAM components on the same SAP or SDP binding; if the MEP is a down MEP, the fault is propagated to the OAM components on the mate SAP or SDP-binding at the other side of the service.

#### 3.5.4.1 SAP and SDP-binding failure (including pseudowire status)

When a SAP or SDP-binding becomes faulty (oper-down, admin-down, or pseudowire status faulty), SMGR needs to propagate the fault to up MEPs on the same SAP or SDP-bindings about the fault, as well as to OAM components (such as down MEPs and E-LMI) on the mate SAP or SDP-binding.

#### 3.5.4.2 Service down

This section describes procedures for the scenario where an Epipe service is down because of the following:

- Service is administratively shutdown. When service is administratively shutdown, the fault is propagated to the SAP and SDP-bindings in the service.



- If the Epipe service is used as a PBB tunnel into a B-VPLS, the Epipe service is also considered operationally down when the B-VPLS service is administratively shutdown or operationally down. If this is the case, fault is propagated to the Epipe SAP.
- In addition, one or more SAPs or SDP-bindings in the B-VPLS can be configured to propagate fault to this Epipe (see fault-propagation-bmac below). If the B-VPLS is operationally up but all of these entities have detected fault or are down, the fault is propagated to this Epipe's SAP.

### 3.5.4.3 Interaction with pseudowire redundancy

When a fault occurs on the SAP side, the pseudowire status bit is set for both active and standby pseudowires. When only one of the pseudowire is faulty, SMGR does not notify CFM. The notification occurs only when both pseudowire becomes faulty. The SMGR propagates the fault to CFM.

Because there is no fault handling in the pipe service, any CFM fault detected on an SDP binding is not used in the pseudowire redundancy's algorithm to choose the most suitable SDP binding to transmit on.

## 3.5.5 Ipipe services

### 3.5.5.1 SAP or SDP-binding failure (including pseudowire status)

When a SAP or SDP-binding becomes faulty (oper-down, admin-down, or pseudowire status faulty), SMGR propagates the fault to OAM components on the mate SAP or SDP-binding.

### 3.5.5.2 Service administratively shutdown

When the service is administratively shutdown, SMGR propagates the fault to OAM components on both SAP or SDP-bindings.

### 3.5.5.3 Interaction with pseudowire redundancy

When the fault occurs on the SAP side, the pseudowire status bit is set for both active and standby pseudowires.

When only one of the pseudowire is faulty, SMGR does not notify CFM. The notification only occurs when both pseudowires become faulty. Then the SMGR propagates the fault to CFM. Because there is no fault handling in the pipe service, any CFM fault detected on a SDP-binding is not used in the pseudowire redundancy's algorithm to choose the most suitable SDP-binding to transmit on.

## 3.5.6 VPLS service

For VPLS services, on down MEPs are supported for fault propagation.

### 3.5.6.1 CFM detected fault

When a MEP detects a fault and fault propagation is enabled for the MEP, CFM communicates the fault to the SMGR. The SMGR marks the SAP and SDP-binding as oper-down. Note that oper-down is used here in VPLS instead of "oper-up but faulty" in the pipe services. CFM traffic can be transmitted to or received from the SAP and SDP-binding to ensure when the fault is cleared, the SAP goes back to normal operational state.

Note that as stated in [CFM connectivity fault conditions](#), a fault is raised whenever a remote MEP is down (not all remote MEPs have to be down). When it is not desirable to trigger fault handling actions in some cases when a down MEP has multiple remote MEPs, operators can disable fault propagation for the MEP.

If the MEP is a down MEP, SMGR performs the fault handling actions for the affected services. Local actions done by the SMGR include (but are not limited to):

- Flushing MAC addresses learned on the faulty SAP and SDP-binding.
- Triggering transmission of MAC flush messages.
- Notifying MSTP/RSTP about topology change. If the VPLS instance is a management VPLS (mVPLS), all VPLS instances that are managed by the m VPLS inherits the MSTP/RSTP state change and react accordingly to it.
- If the service instance is a B-VPLS, and fault-propagation-bmac address(es) is/are configured for the SAP and SDP-binding, SMGR performs a lookup using the B-MAC address(es) to find out which pipe services need to be notified, then propagates a fault to these services. There can be up to four remote B-MAC addresses associated with an SAP and SDP-binding for the same B-VPLS.

### 3.5.6.2 SAP and SDP-binding failure (including pseudowire status)

If the service instance is a B-VPLS, and an associated B-MAC address is configured for the failed SAP and SDP-binding, the SMGR performs a lookup using the B-MAC address to find out which pipe services are notified and then propagate fault to these services.

Within the same B-VPLS service, all SAPs/SDP-bindings configured with the same fault propagation B-MACs must be faulty or oper down for the fault to be propagated to the appropriate pipe services.

### 3.5.6.3 Service down

When a VPLS service is down:

- If the service is not a B-VPLS service, the SMGR propagates the fault to OAM components on all SAP and SDP-bindings in the service.
- If the service is a B-VPLS service, the SMGR propagates the fault to OAM components on all SAP and SDP-bindings in the service as well as all pipe services that are associated with the B-VPLS instance.

### 3.5.6.4 Pseudowire redundancy and Spanning Tree Protocol

A SAP or SDP binding that has a down MEP fault is made operationally down. This causes pseudowire redundancy or Spanning Tree Protocol (STP) to take the appropriate actions.

However, the reverse is not true. If the SAP or SDP binding is blocked by STP, or is not tx-active because of pseudowire redundancy, no fault is generated for this entity.

### 3.5.7 IES and VPRN services

For IES and VPRN services, only down MEP is supported on Ethernet SAPs and spoke SDP bindings.

When a down MEP detects a fault and fault propagation is enabled for the MEP, CFM communicates the fault to the SMGR. The SMGR marks the SAP/SDP binding as operationally down. CFM traffic can still be transmitted to or received from the SAP and SDP-binding to ensure when the fault is cleared and the SAP goes back to normal operational state.

Because the SAP and SDP-binding goes down, it is not usable to upper applications. In this case, the IP interface on the SAP and SDP-binding go down. The prefix is withdrawn from routing updates to the remote PEs. The same applies to subscriber group interface SAPs on the 7450 ESS and 7750 SR.

When the IP interface is administratively shutdown, the SMGR notifies the down MEP and a CFM fault notification is generated to the CPE through interface status TLV or suspension of CCM based on local configuration.

### 3.5.8 Pseudowire switching

When the node acts as a pseudowire switching node, meaning two pseudowires are stitched together at the node, the SMGR does not communicate pseudowire failures to CFM. Such features are expected to be communicated by pseudowire status messages, and CFM runs end-to-end on the head end and tail end of the stitched pseudowire for failure notification.

### 3.5.9 LLF and CFM fault propagation

LLF and CFM fault propagation are mutually exclusive. CLI protection is in place to prevent enabling both LLF and CFM fault propagation in the same service, on the same node and at the same time. However, there are still instances where irresolvable fault loops can occur when the two schemes are deployed within the same service on different nodes. This is not preventable by the CLI. At no time should these two fault propagation schemes be enabled within the same service.

### 3.5.10 802.3ah EFM OAM mapping and interaction with service manager

802.3ah EFM OAM declares a link fault when any of the following occurs:

- loss of OAMPDU for a specific period of time
- receiving OAMPDU with link fault flags from the peer

When 802.3ah EFM OAM declares a fault, the port goes into operation state down. The SMGR communicates the fault to CFM MEPs in the service.

OAM fault propagation in the opposite direction (SMGR to EFM OAM) is not supported.

## 3.6 Bidirectional Forwarding Detection

Bidirectional Forwarding Detection (BFD) is an efficient, short-duration detection of failures in the path between two systems. If a system stops receiving BFD messages for a long enough period (based on

configuration), it is assumed that a failure along the path has occurred and the associated protocol or service is notified of the failure.

BFD can provide a mechanism used for failure detection over any media, at any protocol layer, with a wide range of detection times and overhead, to avoid a proliferation of different methods.

SR OS supports asynchronous and on-demand modes of BFD in which BFD messages are sent to test the path between systems.

If multiple protocols are running between the same two BFD endpoints, only a single BFD session is established, and all associated protocols share the single BFD session.

As well as the typical asynchronous mode, there is also an echo function defined within RFC 5880, *Bidirectional Forwarding Detection*, that allows either of the two systems to send a sequence of BFD echo packets to the other system, which loops them back within that system’s forwarding plane. If a number of these echo packets are lost, the BFD session is declared down.

### 3.6.1 BFD control packet

The base BFD specification does not specify the encapsulation type to be used for sending BFD control packets. Instead, use the appropriate encapsulation type for the medium and network. The encapsulation for BFD over IPv4 and IPv6 networks is specified in RFC 5881, *Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)*, and RFC 5883, *Bidirectional Forwarding Detection (BFD) for Multihop Paths*, BFD for IPv4 and IPv6. This specification requires that BFD control packets be sent over UDP with a destination port number of 3784 (single hop) or 4784 (multi-hop paths) and the source port number must be within the range 49152 to 65535.

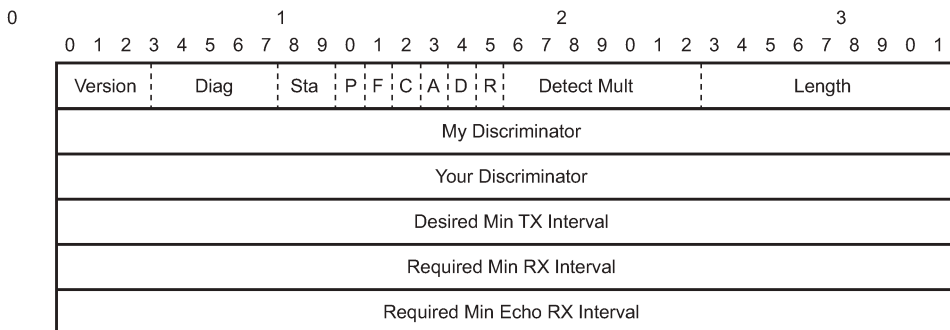
Also, the TTL of all transmitted BFD packets must have an IP TTL of 255. All BFD packets received must have an IP TTL of 255 if authentication is not enabled. If authentication is enabled, the IP TTL should be 255, but can still be processed if it is not (assuming the packet passes the enabled authentication mechanism).

If multiple BFD sessions exist between two nodes, the BFD discriminator is used to de-multiplex the BFD control packet to the appropriate BFD session.

### 3.6.2 Control packet format

The BFD control packet has two sections: a mandatory section and an optional authentication section.

Figure 55: Mandatory frame format



al\_0893

Table 15: BFD control packet field descriptions

Field	Description
Vers	The version number of the protocol. The initial protocol version is 0.
Diag	A diagnostic code specifying the local system's reason for the last transition of the session from Up to some other state. Possible values are: 0-No diagnostic 1-Control detection time expired 2-Echo function failed 3-Neighbor signaled session down 4-Forwarding plane reset 5-Path down 6-Concatenated path down 7-Administratively down
D Bit	The demand mode bit. (Not supported)
P Bit	The poll bit. If set, the transmitting system is requesting verification of connectivity, or of a parameter change.
F Bit	The final bit. If set, the transmitting system is responding to a received BFD control packet that had the poll (P) bit set.
Rsvd	Reserved bits. These bits must be zero on transmit and ignored on receipt.
Length	Length of the BFD control packet, in bytes.
My Discriminator	A unique, non-zero discriminator value generated by the transmitting system, used to demultiplex multiple BFD sessions between the same pair of systems.
Your Discriminator	The discriminator received from the corresponding remote system. This field reflects back the received value of my discriminator, or is zero if that value is unknown.
Desired Min TX Interval	This is the minimum interval, in microseconds, that the local system would like to use when transmitting BFD control packets.
Required Min RX Interval	This is the minimum interval, in microseconds, between received BFD control packets that this system is capable of supporting.
Required Min Echo RX Interval	This is the minimum interval, in microseconds, between received BFD echo packets that this system is capable of supporting. If this value is zero, the transmitting system does not support the receipt of BFD echo packets.

### 3.6.3 Echo support

Echo support for BFD calls for the support of the echo function within BFD. By supporting BFD echo, the router loops back received BFD echo messages to the original sender based on the destination IP address in the packet.

The echo function is useful when the local router does not have sufficient CPU power to handle a periodic polling rate at a high frequency. Therefore, it relies on the echo sender to send a high rate of BFD echo messages through the receiver node, which is only processed by the receiver's forwarding path. This allows the echo sender to send BFD echo packets at any rate.

SR OS does not support the sending of echo requests, only the response to echo requests.

### 3.6.4 Centralized BFD

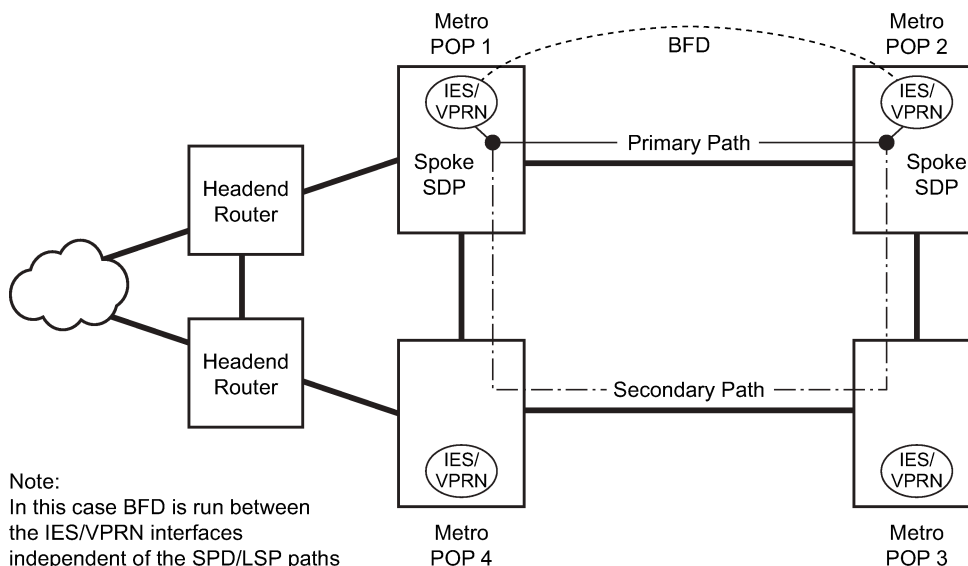
The following applications of centralized BFD require BFD to run on the SF/CPM.

#### 3.6.4.1 IES over spoke SDP

One application for a central BFD implementation is so BFD can be supported over spoke SDPs used to inter-connect IES or VPRN interfaces. When there are spoke SDPs for inter-connections over an MPLS network between two routers, BFD is used to speed up failure detections between nodes so re-convergence of unicast and multicast routing information can begin as quickly as possible.

The MPLS LSP associated with the spoke SDP can enter or egress from multiple interfaces on the router. BFD for these types of interfaces cannot exist on the IOM/XCM by itself.

Figure 56: BFD for IES/VPRN over spoke SDP

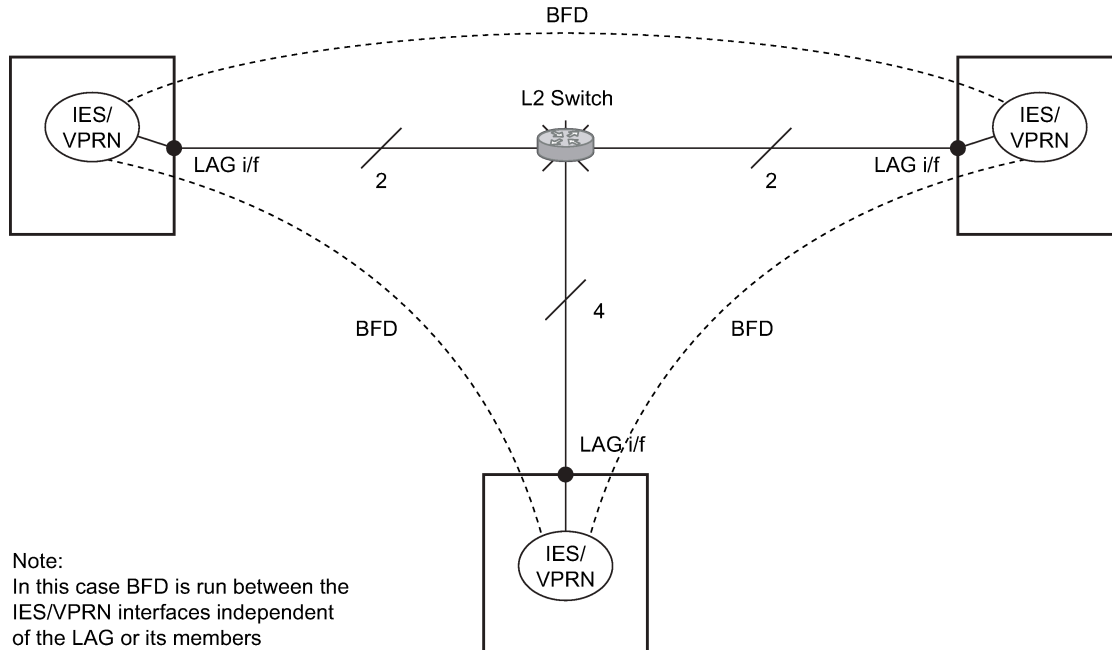


Fig\_31

### 3.6.4.2 BFD over LAG and VSM interfaces

A second application for a central BFD implementation is so BFD can be supported over LAG or VSM interface. This is useful where BFD is not used for link failure detection, but for node failure detection. In this application, the BFD session can run between the IP interfaces associated with the LAG or VSM interface, but there is only one session between the two nodes. There is no requirement for the message flow to across a specific link, or VSM, to get to the remote node.

Figure 57: BFD over LAG and VSM interfaces



Fig\_32

### 3.6.4.3 BFD on an unnumbered IPv4 interface

BFD sessions can be associated with an unnumbered IPv4 interface to monitor the liveness of the connection for IP and MPLS routing protocols, when routing protocol adjacencies and static routes are configured to use this function. When a BFD session is associated with an unnumbered interface as the local anchor point, the BFD parameters are taken from the BFD configuration under the unnumbered interface context. If the BFD parameters are not configured within the unnumbered interface context, then BFD sessions are not attempted. All BFD sessions associated with an unnumbered interface are automatically run on the FP complex associated with the CPM.

### 3.6.4.4 LSP BFD and VCCV BFD

BFD is supported over MPLS-TP, RSVP, and LDP LSPs, as well as over pseudowires that support Layer 2 services such as Epipe VPLS spoke-SDPs and mesh-SDPs using centralized BFD. See the 7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide and 7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 2 Services and EVPN Guide for more information.

### 3.6.4.5 Seamless BFD for SR-TE LSPs

For more information, see the [Seamless Bidirectional Forwarding Detection](#).

## 3.6.5 Seamless Bidirectional Forwarding Detection

Seamless BFD (S-BFD), RFC 7880, *Seamless Bidirectional Forwarding Detection (S-BFD)*, is a form of BFD that avoids the negotiation and state establishment for the BFD sessions. This is done primarily by pre-determining the session discriminator and then using other mechanisms to distribute the discriminators to a remote network entity. This allows client applications or protocols to more quickly initiate and perform connectivity tests. Furthermore, a per-session state is maintained only at the head end of a session. The tail end simply reflects BFD control packets back to the head end.

A seamless BFD session is established between an initiator and a reflector. There is only one instance of a reflector per SR OS router. A discriminator is assigned to the reflector. Each of the initiators on a router is also assigned a discriminator.

By default, S-BFD operates in asynchronous mode where the reflector encapsulates and routes IP/UDP encapsulated S-BFD packets back to the initiator using the IGP shortest path. However, some applications also support a controlled return TE path for S-BFD reply packets, where S-BFD operates in echo mode and the reflector router forwards packets back toward the initiator on a specified labelled path using, for example, an SR policy. For more information, see the application-specific descriptions for S-BFD in the *7750 SR and 7950 XRS Segment Routing and PCE User Guide*.

Seamless BFD sessions are created on the request of a client application, for example, MPLS. This user guide describes the base S-BFD configuration required on initiator and reflector routers. Application-specific configuration is required to create S-BFD sessions.

### 3.6.5.1 S-BFD reflector configuration and behavior

The S-BFD reflector is configured by using the following CLI:

```
configure
  bfd
    seamless-bfd
      [no] reflector <name>
        discriminator <value>
        description <string>
        local-state up | admin-down
        [no] shutdown
```

The discriminator value must be allocated from the S-BFD reflector pool, 524288 - 526335. When the router receives an S-BFD packet from the initiator, with the local router's S-BFD discriminator as the "YourDiscriminator" field, then the local node sends the S-BFD packet back to the initiator via a routed path. The state field in the reflected packet is populated with either the Up or AdminDown state value based on the local-state configuration.



**Note:** Only a single reflector discriminator per node is supported, and the reflector cannot be no shutdown unless at least a discriminator is configured.

Seamless BFD control packets are discarded when the reflector is not configured, is shutdown, or the "YourDiscriminator" field does not match the discriminator of the reflector. Both IPv4 and IPv6 are



supported, but in the case of IPv6, the reflector can only reflect BFD control packets with a global unicast destination address.

### 3.6.5.2 S-BFD initiator global configuration

Before an application can request the establishment of an S-BFD session, a mapping table of remote discriminators to peer far-end IP addresses must exist. The mapping can be accomplished in two ways:

- statically configured
- automatically learned using opaque OSPF/IS-IS routing extensions

See [Static S-BFD discriminator configuration](#) and [Automated S-BFD discriminator distribution](#) for more information about mapping remote discriminators to IP-addresses and to originated router-id.

#### 3.6.5.2.1 Static S-BFD discriminator configuration

To statically map a Seamless BFD remote IP address with its discriminator, use the following CLI commands:

```
config>router>bfd
  seamless-bfd
    peer <ip-address> discriminator <remote-discriminator>
    peer <ip-address> discriminator <remote-discriminator>
    ...
  exit
```

The S-BFD initiator immediately starts sending S-BFD packets if the discriminator value of the far-end reflector is known, no session setup is required.

With S-BFD sessions, there is no INIT state. The initiator state changes from AdminDown to Up when it begins to send (initiate) S-BFD packets.

The S-BFD initiator sends the BFD packet to the reflector using the following fields:

#### **Src IP**

This field contains the local session IP address. For IPv6, this is a global unicast address belonging to the node.

#### **Dst IP**

This field contains the reflector's IP address (configured).

#### **MyDiscriminator**

This field contains the locally assigned discriminator.

#### **YourDiscriminator**

This field contains the reflector's discriminator value.

If the initiator receives a valid response from the reflector with an Up state, the initiator declares the S-BFD session state as Up.

If the initiator fails to receive a specific number of responses, as determined by the BFD multiplier in the BFD template for the session, the initiator declares the S-BFD session state as Failed.

If any of the discriminators change, the session fails and the router attempts to restart with the new values. If the reflector discriminator changes at the far-end peer, the session fails, but the mapping may not have been updated locally before the system checks for a new reflector discriminator from the local mapping

table. The session is bounced, bringing it up with the new values. If any discriminator is deleted, the corresponding S-BFD sessions are deleted.

### 3.6.5.2.2 Automated S-BFD discriminator distribution

It is possible to automatically map an S-BFD remote IP address with its discriminator using IGP routing protocol extensions. The required protocol extensions are introduced by RFC 7883 for IS-IS and RFC 7884 for OSPF. These extensions provide the encodings to advertise the S-BFD discriminators as opaque information within the advertised IGP link state information. BGP-LS added extensions allow the export of IS-IS and OSPF S-BFD discriminator information using encodings defined in *draft-ietf-idr-bgp-ls-sbfd-extensions-01*.

Two preconditions must apply before automated mapping of S-BFD discriminators is enabled:

#### traffic-engineering

This enables the TE-DB infrastructure to create the mapping between an IP address and an S-BFD discriminator.



**Note:** **traffic-engineering** is not supported in VPRN or for OSPFv3.

#### advertise-router-capability

This encodes the S-BFD discriminator in OSPF or IS-IS opaque Router Information TLVs.

The following is an example of an OSPF configuration output:

```
A:Router-A>config>bfd# info detail
  seamless-bfd
    reflector "aaa"
      no description
      discriminator 525002
      local-state up
      no shutdown
    exit
  exit
```

```
A:Router-A>config>router>ospf# info
-----
  router-id 10.20.1.1
  traffic-engineering
  advertise-router-capability area
  area 0.0.0.0
    interface "system"
      no shutdown
    exit
  interface "to_Router-C"
    hello-interval 2
    dead-interval 10
    metric 1000
    no shutdown
  exit
  interface "to_Router-B"
    hello-interval 2
    dead-interval 10
    metric 1000
    no shutdown
  exit
  exit
```

```
no shutdown
-----
*A:Dut-A>config>router>ospf#
```

### 3.7 Traceroute with ICMP tunneling in common applications

This section provides example output of the traceroute OAM tool when the ICMP tunneling feature is enabled in a few common applications.

The ICMP tunneling feature is described in [Tunneling of ICMP reply packets over MPLS LSP](#) and provides supports for appending to the ICMP reply of type Time Exceeded the MPLS label stack object defined in RFC 4950. The new MPLS Label Stack object allows an LSR to include label stack information including label value, EXP, and TTL field values, from the encapsulation header of the packet that expired at the LSR node.

#### 3.7.1 BGP-LDP stitching and ASBR/ABR/data path RR for BGP IPv4 label route

```
ASBR1      ASBR2
----- D ----- E -----
|          |          |          |
A ----- C          F ----- B
DSLAM1    PE1        PE2      DSLAM2
|          |          |          |
----- G ----- H -----
          ASBR3     ASBR4

# lsp-trace ldp-bgp stitching
*A:Dut-A# oam lsp-trace prefix 10.20.1.6/32 detail downstream-map-
tlv dmap
lsp-trace to 10.20.1.6/32: 0 hops min, 0 hops max, 108 byte packets
1 10.20.1.1 rtt=2.89ms rc=15(LabelSwitchedWithFecChange) rsc=1
   DS 1: ipaddr=10.10.1.2 ifaddr=10.10.1.2 iftype=ipv4Numbered MRU=1496
        label[1]=262143 protocol=3(LDP)
        label[2]=262139 protocol=2(BGP)
        fecchange[1]=POP fectype=LDP IPv4 prefix=10.20.1.6 remotepeer=0.0.0.0 (Unknown)
        fecchange[2]=PUSH fectype=BGP IPv4 prefix=10.20.1.6 remotepeer=10.20.1.2
        fecchange[3]=PUSH fectype=LDP IPv4 prefix=10.20.1.2 remotepeer=10.10.1.2
2 10.20.1.2 rtt=5.19ms rc=3(EgressRtr) rsc=2
2 10.20.1.2 rtt=5.66ms rc=8(DSRtrMatchLabel) rsc=1
   DS 1: ipaddr=10.10.4.4 ifaddr=10.10.4.4 iftype=ipv4Numbered MRU=0
        label[1]=262138 protocol=2(BGP)
3 10.20.1.4 rtt=6.53ms rc=15(LabelSwitchedWithFecChange) rsc=1
   DS 1: ipaddr=10.10.6.5 ifaddr=10.10.6.5 iftype=ipv4Numbered MRU=1496
        label[1]=262143 protocol=3(LDP)
        label[2]=262138 protocol=2(BGP)
        fecchange[1]=PUSH fectype=LDP IPv4 prefix=10.20.1.5 remotepeer=10.10.6.5
4 10.20.1.5 rtt=8.51ms rc=3(EgressRtr) rsc=2
4 10.20.1.5 rtt=8.45ms rc=15(LabelSwitchedWithFecChange) rsc=1
   DS 1: ipaddr=10.10.10.6 ifaddr=10.10.10.6 iftype=ipv4Numbered MRU=1496
        label[1]=262143 protocol=3(LDP)
        fecchange[1]=POP fectype=BGP IPv4 prefix=10.20.1.6 remotepeer=0.0.0.0
(Unknown)
        fecchange[2]=PUSH fectype=LDP IPv4 prefix=10.20.1.6 remotepeer=10.10.10.6
```

```
5 10.20.1.6 rtt=11.2ms rc=3(EgressRtr) rsc=1
```

```
*A:Dut-A# configure router ldp-  
shortcut (to add ldp label on first hop but overall behavior is similar)
```

```
# 12.0R4 default behavior (we have routes back to the source)
```

```
*A:Dut-A# traceroute 10.20.1.6 detail wait 100  
traceroute to 10.20.1.6, 30 hops max, 40 byte packets  
 1  1 10.10.2.1 (10.10.2.1) 3.47 ms  
 1  2 10.10.2.1 (10.10.2.1) 3.65 ms  
 1  3 10.10.2.1 (10.10.2.1) 3.46 ms  
 2  1 10.10.1.2 (10.10.1.2) 5.46 ms  
 2  2 10.10.1.2 (10.10.1.2) 5.83 ms  
 2  3 10.10.1.2 (10.10.1.2) 5.20 ms  
 3  1 10.10.4.4 (10.10.4.4) 8.55 ms  
 3  2 10.10.4.4 (10.10.4.4) 7.45 ms  
 3  3 10.10.4.4 (10.10.4.4) 7.29 ms  
 4  1 10.10.6.5 (10.10.6.5) 9.67 ms  
 4  2 10.10.6.5 (10.10.6.5) 10.1 ms  
 4  3 10.10.6.5 (10.10.6.5) 10.9 ms  
 5  1 10.20.1.6 (10.20.1.6) 11.5 ms  
 5  2 10.20.1.6 (10.20.1.6) 11.1 ms  
 5  3 10.20.1.6 (10.20.1.6) 11.4 ms
```

```
# Enable ICMP tunneling on PE and ASBR nodes.
```

```
*A:Dut-D# # configure router ttl-propagate label-route-local all *A:Dut-  
C,D,E,F# configure router icmp-tunneling
```

```
*A:Dut-C# traceroute 10.20.1.6 detail wait 100  
traceroute to 10.20.1.6, 30 hops max, 40 byte packets  
 1  1 10.10.1.1 (10.10.1.1) 11.8 ms  
    returned MPLS Label Stack Object  
    entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1  
 1  2 10.10.1.1 (10.10.1.1) 12.5 ms  
    returned MPLS Label Stack Object  
    entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1  
 1  3 10.10.1.1 (10.10.1.1) 12.9 ms  
    returned MPLS Label Stack Object  
    entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1  
 2  1 10.10.4.2 (10.10.4.2) 13.0 ms  
    returned MPLS Label Stack Object  
    entry 1: MPLS Label = 262143, Exp = 7, TTL = 1, S = 0  
    entry 2: MPLS Label = 262139, Exp = 7, TTL = 1, S = 1  
 2  2 10.10.4.2 (10.10.4.2) 13.0 ms  
    returned MPLS Label Stack Object  
    entry 1: MPLS Label = 262143, Exp = 7, TTL = 1, S = 0  
    entry 2: MPLS Label = 262139, Exp = 7, TTL = 1, S = 1  
 2  3 10.10.4.2 (10.10.4.2) 12.8 ms  
    returned MPLS Label Stack Object  
    entry 1: MPLS Label = 262143, Exp = 7, TTL = 1, S = 0  
    entry 2: MPLS Label = 262139, Exp = 7, TTL = 1, S = 1  
 3  1 10.10.6.4 (10.10.6.4) 10.1 ms  
    returned MPLS Label Stack Object  
    entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1  
 3  2 10.10.6.4 (10.10.6.4) 11.1 ms  
    returned MPLS Label Stack Object  
    entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1  
 3  3 10.10.6.4 (10.10.6.4) 9.70 ms  
    returned MPLS Label Stack Object  
    entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1  
 4  1 10.10.10.5 (10.10.10.5) 12.5 ms  
    returned MPLS Label Stack Object
```

```

        entry 1: MPLS Label = 262143, Exp = 7, TTL = 255, S = 0
        entry 2: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1
4  2  10.10.10.5 (10.10.10.5) 11.9 ms
    returned MPLS Label Stack Object
        entry 1: MPLS Label = 262143, Exp = 7, TTL = 255, S = 0
        entry 2: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1
4  3  10.10.10.5 (10.10.10.5) 11.8 ms
    returned MPLS Label Stack Object
        entry 1: MPLS Label = 262143, Exp = 7, TTL = 255, S = 0
        entry 2: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1
5  1  10.20.1.6 (10.20.1.6) 12.2 ms
5  2  10.20.1.6 (10.20.1.6) 12.5 ms
5  3  10.20.1.6 (10.20.1.6) 13.2 ms

# With lsr-label-route all on all LSRs (only needed on Dut-E) *A:Dut-
E# configure router ttl-propagate lsr-label-route all

*A:Dut-
A# traceroute 10.20.1.6 detail wait 100 traceroute to 10.20.1.6, 30 hops max, 40 byte packets
1  1  10.10.1.1 (10.10.1.1) 12.4 ms
    returned MPLS Label Stack Object
        entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1
1  2  10.10.1.1 (10.10.1.1) 11.9 ms
    returned MPLS Label Stack Object
        entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1
1  3  10.10.1.1 (10.10.1.1) 12.7 ms
    returned MPLS Label Stack Object
        entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1
2  1  10.10.4.2 (10.10.4.2) 11.6 ms
    returned MPLS Label Stack Object
        entry 1: MPLS Label = 262143, Exp = 7, TTL = 1, S = 0
        entry 2: MPLS Label = 262139, Exp = 7, TTL = 1, S = 1
2  2  10.10.4.2 (10.10.4.2) 13.5 ms
    returned MPLS Label Stack Object
        entry 1: MPLS Label = 262143, Exp = 7, TTL = 1, S = 0
        entry 2: MPLS Label = 262139, Exp = 7, TTL = 1, S = 1
2  3  10.10.4.2 (10.10.4.2) 11.9 ms
    returned MPLS Label Stack Object
        entry 1: MPLS Label = 262143, Exp = 7, TTL = 1, S = 0
        entry 2: MPLS Label = 262139, Exp = 7, TTL = 1, S = 1
3  1  10.10.6.4 (10.10.6.4) 9.21 ms
    returned MPLS Label Stack Object
        entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1
3  2  10.10.6.4 (10.10.6.4) 9.58 ms
    returned MPLS Label Stack Object
        entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1
3  3  10.10.6.4 (10.10.6.4) 9.38 ms
    returned MPLS Label Stack Object
        entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1
4  1  10.10.10.5 (10.10.10.5) 12.2 ms
    returned MPLS Label Stack Object
        entry 1: MPLS Label = 262143, Exp = 7, TTL = 1, S = 0
        entry 2: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1
4  2  10.10.10.5 (10.10.10.5) 11.5 ms
    returned MPLS Label Stack Object
        entry 1: MPLS Label = 262143, Exp = 7, TTL = 1, S = 0
        entry 2: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1
4  3  10.10.10.5 (10.10.10.5) 11.5 ms
    returned MPLS Label Stack Object
        entry 1: MPLS Label = 262143, Exp = 7, TTL = 1, S = 0
        entry 2: MPLS Label = 262138, Exp = 7, TTL = 1, S = 1
5  1  10.20.1.6 (10.20.1.6) 11.9 ms

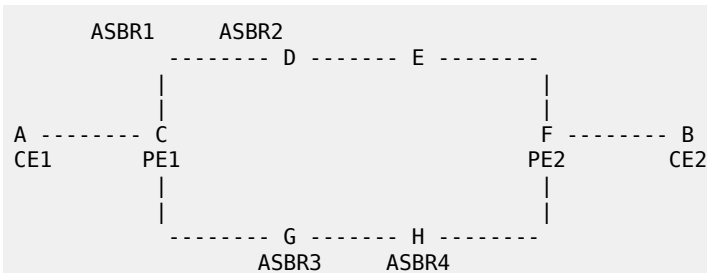
```

```

5 2 10.20.1.6 (10.20.1.6) 12.2 ms
5 3 10.20.1.6 (10.20.1.6) 13.7 ms

```

### 3.7.2 VPRN inter-AS option B



```

# 12.0R4 default behavior (vc-only)
*A:Dut-A# traceroute 3.3.3.4 source 3.3.4.2 wait 100 no-dns
detail traceroute to 3.3.3.4 from 3.3.4.2, 30 hops max, 40 byte packets
 1 1 3.3.4.1 1.97 ms
 1 2 3.3.4.1 1.74 ms
 1 3 3.3.4.1 1.71 ms
 2 1 *
 2 2 *
 2 3 *
 3 1 *
 3 2 *
 3 3 *
 4 1 3.3.3.6 6.76 ms
 4 2 3.3.3.6 7.37 ms
 4 3 3.3.3.6 8.36 ms
 5 1 3.3.3.4 11.1 ms
 5 2 3.3.3.4 9.46 ms
 5 3 3.3.3.4 8.28 ms

```

```

# Configure icmp-tunneling on C, D, E and F

```

```

*A:Dut-A# traceroute 3.3.3.4 source 3.3.4.2 wait 100 no-dns
detail traceroute to 3.3.3.4 from 3.3.4.2, 30 hops max, 40 byte packets
 1 1 3.3.4.1 1.95 ms
 1 2 3.3.4.1 1.85 ms
 1 3 3.3.4.1 1.62 ms
 2 1 10.0.7.3 6.76 ms
    returned MPLS Label Stack Object
      entry 1: MPLS Label = 262143, Exp = 0, TTL = 255, S = 0
      entry 2: MPLS Label = 262140, Exp = 0, TTL = 1, S = 1
 2 2 10.0.7.3 6.92 ms
    returned MPLS Label Stack Object
      entry 1: MPLS Label = 262143, Exp = 0, TTL = 255, S = 0
      entry 2: MPLS Label = 262140, Exp = 0, TTL = 1, S = 1
 2 3 10.0.7.3 7.58 ms
    returned MPLS Label Stack Object
      entry 1: MPLS Label = 262143, Exp = 0, TTL = 255, S = 0
      entry 2: MPLS Label = 262140, Exp = 0, TTL = 1, S = 1
 3 1 10.0.5.4 6.92 ms
    returned MPLS Label Stack Object
      entry 1: MPLS Label = 262140, Exp = 0, TTL = 1, S = 1
 3 2 10.0.5.4 7.03 ms
    returned MPLS Label Stack Object
      entry 1: MPLS Label = 262140, Exp = 0, TTL = 1, S = 1

```

```

3 3 10.0.5.4 8.66 ms
    returned MPLS Label Stack Object
    entry 1: MPLS Label = 262140, Exp = 0, TTL = 1, S = 1
4 1 3.3.3.6 6.67 ms
4 2 3.3.3.6 6.75 ms
4 3 3.3.3.6 6.96 ms
5 1 3.3.3.4 8.32 ms
5 2 3.3.3.4 11.6 ms
5 3 3.3.3.4 8.45 ms

```

```

# With ttl-propagate vprn-transit none on PE1 *A:Dut-C# configure router ttl-
propagate vprn-transit none *A:Dut-B# traceroute 3.3.3.4 source 3.3.4.2 wait 100 no-
dns detail traceroute to 3.3.3.4 from 3.3.4.2, 30 hops max, 40 byte packets

```

```

1 1 3.3.4.1 1.76 ms
1 2 3.3.4.1 1.75 ms
1 3 3.3.4.1 1.76 ms
2 1 3.3.3.6 6.50 ms
2 2 3.3.3.6 6.70 ms
2 3 3.3.3.6 6.36 ms
3 1 3.3.3.4 8.34 ms
3 2 3.3.3.4 7.64 ms
3 3 3.3.3.4 8.73 ms

```

```

# With ttl-propagate vprn-transit all on PE1 *A:Dut-C# configure router ttl-
propagate vprn-transit all *A:Dut-B# traceroute 3.3.3.4 source 3.3.4.2 wait 100 no-
dns detail traceroute to 3.3.3.4 from 3.3.4.2, 30 hops max, 40 byte packets

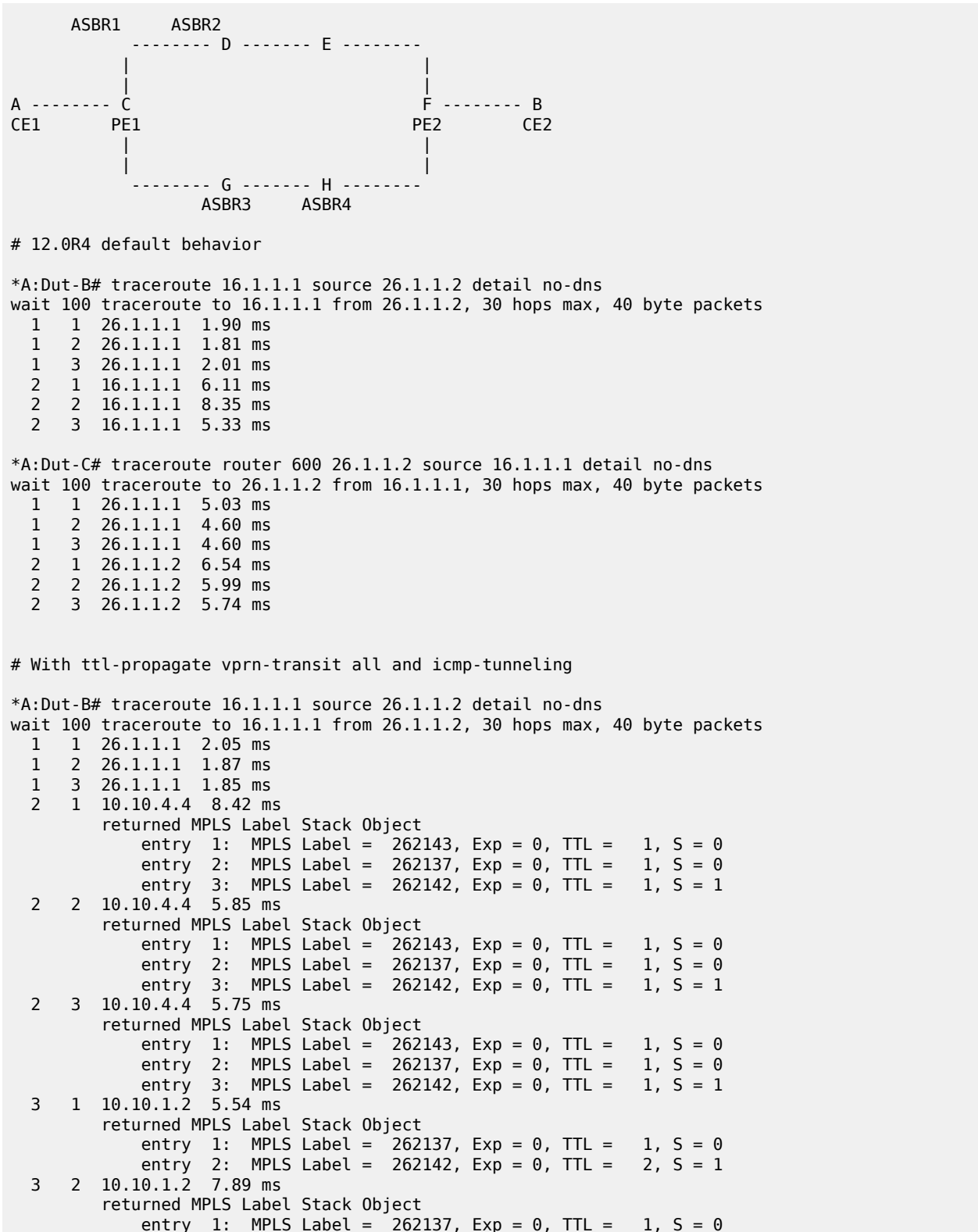
```

```

1 1 3.3.4.1 1.97 ms
1 2 3.3.4.1 1.77 ms
1 3 3.3.4.1 2.37 ms
2 1 10.0.7.3 9.27 ms
    returned MPLS Label Stack Object
    entry 1: MPLS Label = 262143, Exp = 0, TTL = 1, S = 0
    entry 2: MPLS Label = 262140, Exp = 0, TTL = 1, S = 1
2 2 10.0.7.3 6.39 ms
    returned MPLS Label Stack Object
    entry 1: MPLS Label = 262143, Exp = 0, TTL = 1, S = 0
    entry 2: MPLS Label = 262140, Exp = 0, TTL = 1, S = 1
2 3 10.0.7.3 6.19 ms
    returned MPLS Label Stack Object
    entry 1: MPLS Label = 262143, Exp = 0, TTL = 1, S = 0
    entry 2: MPLS Label = 262140, Exp = 0, TTL = 1, S = 1
3 1 10.0.5.4 6.80 ms
    returned MPLS Label Stack Object
    entry 1: MPLS Label = 262140, Exp = 0, TTL = 1, S = 1
3 2 10.0.5.4 6.71 ms
    returned MPLS Label Stack Object
    entry 1: MPLS Label = 262140, Exp = 0, TTL = 1, S = 1
3 3 10.0.5.4 6.58 ms
    returned MPLS Label Stack Object
    entry 1: MPLS Label = 262140, Exp = 0, TTL = 1, S = 1
4 1 3.3.3.6 6.47 ms
4 2 3.3.3.6 6.75 ms
4 3 3.3.3.6 9.06 ms
5 1 3.3.3.4 7.99 ms
5 2 3.3.3.4 9.31 ms
5 3 3.3.3.4 8.13 ms

```

### 3.7.3 VPRN inter-AS option C and ASBR/ABR/data path RR for BGP IPv4 label route





```

    entry 2: MPLS Label = 262142, Exp = 0, TTL = 2, S = 1
3 3 10.10.1.2 5.56 ms
    returned MPLS Label Stack Object
    entry 1: MPLS Label = 262137, Exp = 0, TTL = 1, S = 0
    entry 2: MPLS Label = 262142, Exp = 0, TTL = 2, S = 1
4 1 16.1.1.1 9.50 ms
4 2 16.1.1.1 5.91 ms
4 3 16.1.1.1 5.85 ms

# With ttl-propagate vprn-local all
*A:Dut-C# traceroute router 600 26.1.1.2 source 16.1.1.1 detail no-dns
wait 100 traceroute to 26.1.1.2 from 16.1.1.1, 30 hops max, 40 byte packets
1 1 10.10.4.2 4.78 ms
    returned MPLS Label Stack Object
    entry 1: MPLS Label = 262143, Exp = 7, TTL = 1, S = 0
    entry 2: MPLS Label = 262136, Exp = 7, TTL = 1, S = 0
    entry 3: MPLS Label = 262142, Exp = 7, TTL = 1, S = 1
1 2 10.10.4.2 4.56 ms
    returned MPLS Label Stack Object
    entry 1: MPLS Label = 262143, Exp = 7, TTL = 1, S = 0
    entry 2: MPLS Label = 262136, Exp = 7, TTL = 1, S = 0
    entry 3: MPLS Label = 262142, Exp = 7, TTL = 1, S = 1
1 3 10.10.4.2 4.59 ms
    returned MPLS Label Stack Object
    entry 1: MPLS Label = 262143, Exp = 7, TTL = 1, S = 0
    entry 2: MPLS Label = 262136, Exp = 7, TTL = 1, S = 0
    entry 3: MPLS Label = 262142, Exp = 7, TTL = 1, S = 1
2 1 10.10.6.4 4.55 ms
    returned MPLS Label Stack Object
    entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 0
    entry 2: MPLS Label = 262142, Exp = 7, TTL = 2, S = 1
2 2 10.10.6.4 4.47 ms
    returned MPLS Label Stack Object
    entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 0
    entry 2: MPLS Label = 262142, Exp = 7, TTL = 2, S = 1
2 3 10.10.6.4 4.20 ms
    returned MPLS Label Stack Object
    entry 1: MPLS Label = 262138, Exp = 7, TTL = 1, S = 0
    entry 2: MPLS Label = 262142, Exp = 7, TTL = 2, S = 1
3 1 26.1.1.1 4.62 ms
3 2 26.1.1.1 4.41 ms
3 3 26.1.1.1 4.64 ms
4 1 26.1.1.2 5.74 ms
4 2 26.1.1.2 6.22 ms
4 3 26.1.1.2 5.77 ms

```

### 3.8 Hashing visibility tool

The hashing visibility tool allows users to define a test packet and then inject that packet into a specified ingress port. The result of the test displays the egress port, routing context, egress interface name, and the IP next hop used to forward the packet.

There are three major steps when running this test:

1. Configure header templates needed to build test packets.
2. Configure parameter overrides and build packet header sequences.
3. Execute the **find-egress** test specifying the ingress port.

Execute the test with the **oam find-egress packet** *packet-number ingress-port port-id* command. This causes the specified test frame or packet to be injected at the specified port and report the results.

The **find-egress** command is supported on IPv4 and IPv6 routing, Layer 3, Layer 2 VPLS, and Epipe services.

The following lists the supported packet header sequences.

- Ethernet>Payload Ethernet>MPLS>IPv4>GRE>IPv4>Payload
- Ethernet>IPv4>Payload Ethernet>MPLS>Control-Word>Ethernet>Payload
- Ethernet>IPv4>IPSec>Payload Ethernet>MPLS>Control-Word>Ethernet>IPv4>Payload
- Ethernet>IPv4>UDP>Payload Ethernet>MPLS>Control-Word>Ethernet>IPv4>UDP>Payload
- Ethernet>IPv4>UDP>GTP-U>Payload Ethernet>MPLS>Control-Word>Ethernet>IPv4>TCP>Payload
- Ethernet>IPv4>UDP>IPSec>Payload Ethernet>MPLS>Control-Word>Ethernet>IPv6>Payload
- Ethernet>IPv4>UDP>L2TP>Payload Ethernet>MPLS>Control-Word>Ethernet>IPv6>UDP>Payload
- Ethernet>IPv4>TCP>Payload Ethernet>MPLS>Control-Word>Ethernet>IPv6>TCP>Payload
- Ethernet>IPv4>GRE>IPv4>Payload Ethernet>MPLS>IPv4>IPSec>Payload
- Ethernet>IPv6>Payload Ethernet>MPLS>Ethernet>IPv4>IPSec>Payload
- Ethernet>IPv6>UDP>Payload Ethernet>MPLS>Control-Word>Ethernet>IPv4>IPSec>Payload
- Ethernet>IPv6>TCP>Payload Ethernet>MPLS>IPv6>IPSec>Payload
- Ethernet>IPv6>IPSec>Payload Ethernet>MPLS>Ethernet>IPv6>IPSec>Payload
- Ethernet>IPv6>UDP>IPSec>Payload Ethernet>MPLS>Control-Word>Ethernet>IPv6>IPSec>Payload
- Ethernet>IPv6>UDP>GTP-U>Payload Ethernet>MPLS>IPv4>UDP>IPSec>Payload
- Ethernet>IPv6>UDP>L2TP>Payload Ethernet>MPLS>IPv6>UDP>IPSec>Payload
- Ethernet>MPLS>Payload Ethernet>MPLS>Ethernet>IPv4>UDP>IPSec>Payload
- Ethernet>MPLS>Control-Word>Payload Ethernet>MPLS>Ethernet>IPv6>UDP>IPSec>Payload

### 3.8.1 Configuring the header templates

#### About this task

Follow this procedure to configure the header templates:

#### Procedure

**Step 1.** Configure the header in the **configure test-oam build-packet** context.

**Step 2.** Define the possible header types that are to be used in **find-egress** tests.

Any header that needs to be used in the test packet must be created in this step.

The operator can optionally specify a default value for the associated header parameters, however all parameters can also be set or overridden in step 2 of using the [Hashing visibility tool](#).

Only header parameters that are used in the hashing decision-making can be configured. All other parameters are set to valid default values internally.

```
A:node-2>config>test-oam>build-packet
  header 1 create
    ethernet
```

```

src-mac-address AA:BB:CC:DD:EE:FF
dst-mac-address FF:EE:DD:CC:BB:AA
header 2 create
  udp
    src-port 12345
    dest-port 54321
header 3 create
  ipv4
    dscp 0
    src-ipv4-address 11.22.33.44
    dst-ipv4-address 55.66.77.88

```

## 3.8.2 Configuring parameter overrides and header sequences

### About this task

The steps to configure parameter overrides and header sequences are:

### Procedure

**Step 1.** Define the header parameter overrides in the **debug oam build-packet packet field-override** context.

Any header value can be overridden.

**Step 2.** Define the test packet header sequence in the **debug oam build-packet packet header-sequence** context.

The **header-sequence** command includes a string that specifies the sequence of header to create the test packet.

Each header is defined in the form of h<header-number> with a "/" separating the header identifiers.

The headers sequence is defined from outer header to innermost header.

```

A:node-2>debug>oam>build-packet>packet <pkt-id>
field-override
  header 1
    ethernet
      src-mac-address 11:22:33:44:55:66
      dst-mac-address 22:33:44:55:66:77
header-sequence "h1/h3/h2"

```

### What to do next

Execute the test with the **oam find-egress packet *packet-id* ingress-port *port-id*** command. This causes the specified test frame or packet to be injected at specified port and this reports the result.

```

A:bkvm14# oam find-egress packet 1 ingress-port 1/5/7
-----
Egress Information for Packet 1, Ingress Port 1/5/7
-----
Port       : 1/5/1
Router Name : Base
Interface Nm: toDUT-2917
Next Hop   : 10.10.30.2
-----
Test completed.

```

## 4 OAM monitoring and reporting

Several OAM fault and performance tools have been developed to monitor and report information about the network infrastructure and the services that rely on that infrastructure. Most technology-specific tools are categorized under one or more of the following scheduling and reporting functions:

- **Link measurement**

This function performs IP delay measurement scheduling with direct reporting to the routing engine. This function influences the local routing sub system.

- **OAM Performance Monitoring (OAM-PM)**

This function is an Ethernet, IP, and MPLS performance measurement architecture with scheduling, reporting, and delay streaming options. It is focused on northbound system collection.

- **Service Assurance Agent (SAA)**

This function is an Ethernet, IP, and MPLS fault and performance measurement architecture with scheduling and reporting. It is focused on northbound system collection.

The link measurement and OAM-PM functions share common infrastructure. Both functions support the configuration of a source UDP port. This range of source UDP ports allocated to the TWAMP Light is 64374 to 64383. By default, link measurement and OAM-PM use dynamic source UDP ports. However, a specific source UDP port can be configured. Use the following command to configuration to allocate a source UDP port to the application that is going to use the specific port.

```
configure test-oam twamp twamp-light source-udp-port-pools port
```

To change a port allocation from one application to another, the port must not be configured under the application that currently owns the source UDP port. A source UDP port that is configured under the link measurement template or OAM-PM test session cannot be reassigned regardless of the administrative state.



**Note:** When the configured UDP source port is used, the IP session is identified using the following:

- tuple source IP
- destination IP
- source UDP port
- destination UDP port

When executing tests between the same source IP, destination IP, and destination UDP port, the source UDP must be different. This means using a different configured source UDP port in the reserved range or allowing automatic source UDP port allocation, which is the default. The automatic assignment of the source UDP ensures uniqueness. Nokia recommends using caution when using configured source UDP ports.

Use the following command to view the configured allocation and use of the source UDP port.

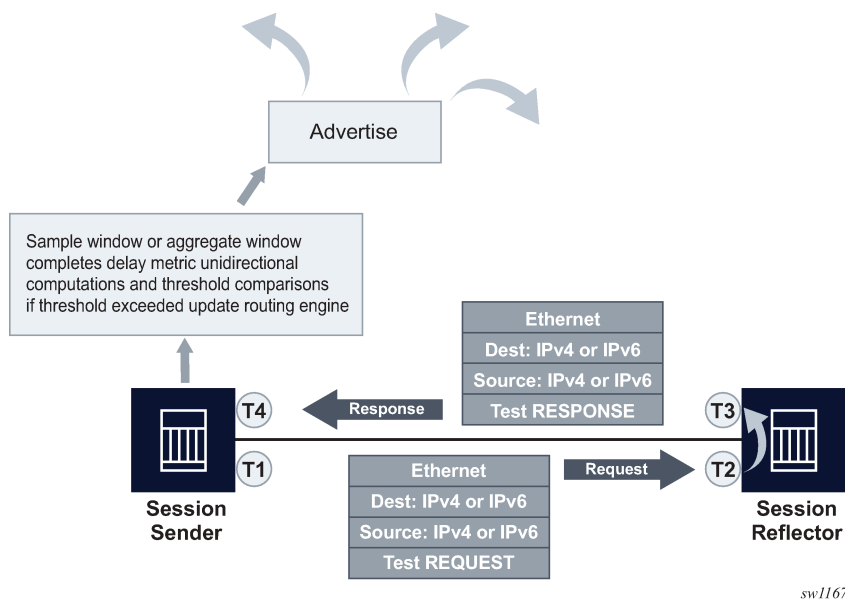
```
show test-oam twamp twamp-light source-udp-port-pools
```

## 4.1 Link measurement

Network elements use routing protocols to exchange information about local links, which can influence routing decisions. These interface attributes are typically static in nature. By using tools specifically designed to measure IP performance, dynamic unidirectional delay can be included in the advertised link attributes.

The following figure shows directly-connected IP interfaces and the link measurement interaction with routing.

Figure 58: Link measurement interactions



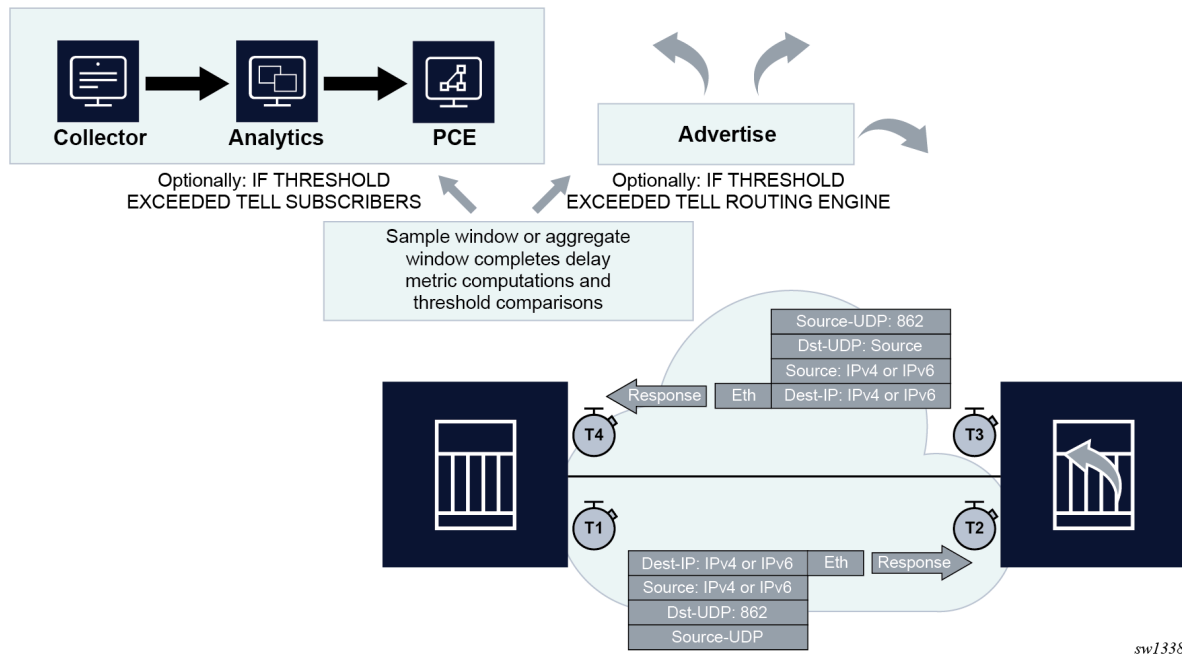
This process can also operate in a reporting disabled mode using the **configure test-oam link-measurement measurement-template reporting** command. In this mode, all processes continue to operate, including threshold comparisons. In the reporting disabled mode, the routing engine is not made aware of the threshold event.

The paths relating to the threshold event, Delay Measure Last Reported (**state test-oam link-measurement router interface last-reported-delay**), Timestamp (**state test-oam link-measurement router interface report-timestamp**), and Triggered By (**state test-oam link-measurement router interface triggered-by**) all support on change notification.

The combination allows the function to support the following notifications for threshold events.

- Inform the routing engine only.
- Inform the routing engine and telemetry subscribers.
- Inform only telemetry subscribers.
- Inform no external functions and maintain results in memory until aged out.

Figure 59: Link measurement interactions with reporting disabled



### 4.1.1 Link measurement template

Common test parameters are included under the `link-measurement>measurement-template template-name` context. After the measurement template has been created, a base router interface can reference the measurement template using the `config>router>interface>if-attribute>delay>dynamic>measurement-template template-name` command. When the association between the interface and the template is established, the interface executes a process to determine the operational state of the test and detect any defect conditions that may prevent proper test execution. Assuming no underlying conditions are present, the IP interface delay measurements are collected in measurement windows compared to configured thresholds, and, when necessary, reported to the routing engine for further processing.

The `measurement-template` context contains the configuration parameters that define the test criteria used by the interface test. Conceptually, the test criteria are divided into the following groups:

- [General configuration](#)
- [Collection and reporting](#)
- [Protocol](#)

#### 4.1.1.1 General configuration

General configuration parameters influence reporting probe frequency, the delay metric type to monitor, and retention of the delay measurement last reported.

The notification process uses the `reporting` command and threshold configurations. By default, reporting is enabled and the routing engine is informed of threshold events. At least one threshold must be configured

to report to the routing engine. Disabling reporting allows the function to execute but prevents the reporting of threshold events to the routing engine. If this value is toggled and previously recorded, the reported value is cleared and the process returns to the initial reporting phase.

The probe frequency, configured using the **interval** command, defines the transmission rate of the test packet.

The **delay** command configures the delay metric (minimum, maximum, or average) that is used for comparison against any configured thresholds. This metric is the same for both types of measurement windows, the **sample-window** and **aggregate-sample-window**.

The **unidirectional-measurement** specifies the method used to compute the unidirectional delay. If the clock synchronization between nodal clocks used by the OAM timestamp function is not synchronized to near exact accuracy, the **derived** option must be used. Specifying this option calculates the unidirectional measurement using the roundtrip delay divided by two computation. If synchronization can meet near exact accuracy, the **actual** option can be used. Specifying this option calculates the forward delay using the forward direction timestamps, T2-T1 computation.

When the operational state of the link measurement test transitions to down, the OAM function instructs the routing engine to clear the last reported delay value at the expiration of the **last-reported-delay-hold**. A previously reported delay is considered valid for the duration of this period and is cleared if the timer reaches zero. If the operational state returns to up before the timer expires, no action is taken to clear the previous value. The counter is reset to the configured value, waiting for the next operational down event.

The operation state for the interface delay test is determined by administrative actions and system events. The administrative events that determine the operational state are:

- administratively disabling the measurement template associated with the interface
- administratively disabling the active IP protocol that is being used to generate the test packet under the **config>router>interface>if-attribute>delay>dynamic-delay** context

The resource issue system events that drive the operational state are:

- unavailability of the UDP port
- internal errors

The aging timer does not start a count to zero for failure conditions that do not affect the interface delay test operational state. The delay measurement last reported is maintained when conditions external to the interface delay test, such as fault conditions on the port, IP interface, routing changes, and so on, occur. If the **last-reported-delay-timer** is set to zero, previously reported delay values from that test are cleared when the operational state changes to down without any additional time.

#### 4.1.1.2 Collection and reporting

The collection and reporting parameters define the length of the **sample-window** and **aggregate-sample-window** and the thresholds that trigger reporting. Two measurement windows are provided to support use cases that require a reporting hierarchy. Both measurement windows include the same configuration options. The threshold values determine when the measurement window is able to update the reported delay value.

The measurement windows use the **multiplier** command to determine the length of time that the measurement window remains open. The sample window length is multiples of the interval. This window stores the results of individual test probes for a total length of the **interval** multiplied by the **multiplier** value. The aggregate sample window multiplier length is the number of sample windows. This window stores the number of results passed from individual sample windows. In the aggregate sample window, the

minimum, maximum, and average calculations are based on the results received from the sample window. For example, if the delay metric of interest is the average, the aggregate sample is a collection of averages passed from the sample window. The reporting in the aggregate sample window is as follows.

- The minimum is the minimum value for all the averages received.
- The maximum is the maximum value from all the averages received.
- The average is the average of all the averages received.

The comparison to thresholds and reporting decisions occurs at the end of the measurement window if it completes without termination and is deemed integral based on the **window-integrity** command configuration. Integrity is a percentage-based calculation that determines the number of samples that must be present in the measurement window for that window to be considered integral. If the number of samples in the window equals or exceeds the number of required samples, the result produced is treated as representative and follows normal post-measurement window processing. However, if the number of samples in the window does not achieve integrity, the result is not considered representative and is only recorded for historical purposes, but is otherwise ignored and not processed. By default, integrity checking is disabled and all results from a measurement window are treated as integral and compared to the configured thresholds.

There are two types of thresholds:

- a microsecond increase or decrease, configured using the **absolute** command
- a percentage increase or decrease, configured using the **relative** command

Thresholding compares the measurement window result to the delay measurement last reported at the end of the successful (completed) measurement window. Reporting is on a per-threshold, per-measurement window basis. If multiple thresholds are reached for a completed measurement window, only one threshold triggers an update to the routing engine. The reporting trigger is recorded with a structure of <Window"Threshold"threshold>. For example, SampleThresholdAbsolute indicates that the absolute threshold in the sample window triggered the report. The report is accompanied by a timestamp and last reported value. When a sample window triggers a new delay measurement, the current aggregate sample window is restarted, preempting any possible reporting from the aggregate sample window. This allows both measurement window types to use the delay measurement last reported as the new benchmark.

Configuration of the measurement windows depends on the specific solution requirements. The two measurement windows collect information regardless of configured thresholds. Both types of measurement windows support their own threshold and integrity configuration. By default, thresholds for both measurement windows are disabled; that is, neither window can report any values to the routing engine.

### 4.1.1.3 Protocol

The protocol parameters influence the format of the test packet, processing, QoS handling, IPv6 discovery, and which influences the return path.

TWAMP Light, also called Simple Two-way Activation Monitoring Protocol (STAMP), is the test packet used to gather IP link measurement delay. The link measurement request source is the session-sender.

TWAMP Light requires the explicit configuration of a reflector on the peer. For SR OS, TWAMP Light reflectors are configured in the **config>router>twamp-light> reflector** context. The reflector is referred to as the session-reflector, the responder to the request.



The session-sender and the session-reflector must agree on the UDP port used to identify TWAMP Light test packets. The SR OS configuration of the session-sender (configured using the **dest-udp-port** command) and session-reflector (configured using the **udp-port** command) must match.

The TWAMP Light test packet was first introduced to support the Network Time Protocol (NTP) encoding of the timestamp in the packet. Updates since the initial standardization of TWAMP Light supports the use of the truncated Precision Time Protocol (PTP) timestamp format. A bit in the TWAMP Light test packet header is repurposed to indicate the timestamp format encoded by the session-sender and the session-reflector. This change leads to some interoperability considerations. The timestamp format should be consistent with the session-sender and session-reflector behavior. The link measurement session-sender can be configured to encode NTP (default) or PTP and set the "z-bit" in the Error Estimate field accordingly. This bit indicates the timestamp format carried in the packet. If the session-reflector sets the "z-bit" in the Error Estimate field to indicate the timestamp format of the reply, the link measurement session-sender can perform the necessary conversion (format and epoch) to produce the correct results. However, if the session-reflector only reflects the original "z-bit" it received from the session-sender and uses a different timestamp format in the packet, the delay calculations are not reliable because of the misinterpretation of the returned timestamp format. SR OS session-reflectors running Release 21.5 and earlier always reflect the "z-bit" received from the session-sender. Regardless of the "z-bit", these session-reflectors always encode an NTP timestamp format in the packet. When these session-reflectors receive a TWAMP Light test packet with the PTP timestamp format, there is a mismatch between the actual timestamp format and the timestamp it has encoded. There is no mechanism for the session-reflector to detect this mismatch and report the correct delay. To ensure accurate delay measurements, any session-sender sending TWAMP Light test packets to an SR OS TWAMP Light reflector that is running Release 21.5 and earlier, must use a timestamp format of NTP. Release 21.7 session-reflectors reply in kind for the timestamp format and properly set the timestamp format bit to match the timestamp encoded by the session-reflector.

IPv6 packets arriving with a UDP checksum of zero are discarded. However, recent work in the IETF is suggesting that selected protocols may register on the local router to accept and process IPv6 packets with a UDP checksum of zero. To provide interoperability, the **allow-ipv6-udp-checksum-zero** command allows the session-sender and the session-reflector to process IPv6 TWAMP Light test packets that arrive with a UDP checksum of zero. This is specific to the link measurement template session-sender and session-reflector and only for the specific UDP ports for TWAMP Light test packets.

IPv6 destination address discovery allows the discovery of a single directly-connected IPv6 peer. When this option is enabled, a bootstrap function using an ICMPv6 echo request with a destination ff02::2 is generated. When the directly-connected peer responds, the link measurement function uses the source address of the ICMPv6 echo response as the destination address for the link measurement test packets. The process has four main components.

- Enabling the functions using **admin-enable**, set to **disabled** by default.
- Configuring the **discovery-interval** value. This is the initial timer used by the discovery process to discover the peer. This interval is used for the duration of the discovery-timer.
- Implementing the discovery phase. If the timer expires or the peer is discovered before the expiration of the **discovery-timer**, the process reverts back to the **update-interval**.
- Implementing the **update-interval**. This is an optional maintenance component of the peer address that runs at a slower rate. This option is not required and can be disabled in environments where the peer address is unlikely to change. If the peer is not discovered during the **discovery-timer** and the **update-interval** is disabled the peer fails to be discovered. Disable and then enable the IPv6 protocol under configure router interface delay dynamic twamp-light ipv6 can restart the discovery process.

By default, the TWAMP light reflection and the base TWAMP Light session-reflectors use routing to return the response packet to the session-sender. There are instances when it may be beneficial to be selective about the IP interface used to return the packet. For example, when multiple tests are executed on different

interfaces between the same pair of nodes, and using non-directly connected interface addresses, and ECMP exists between the two nodes and the **unidirectional-measurement actual** is used. In this case, the **return-path link** command can be configured under the **measurement-template twamp-light** context. This includes the return path TLV and link sub TLV in the test packet. This configuration instructs the session reflector to send the response out the same IP interface on which it was received. The destination IP address for the response packet must be installed in the forwarding table and reachable from that interface. If the routing engine determines that the prefix is not reachable from that interface the response packet is dropped at the reflector.

#### 4.1.1.4 Modifying measurement template configuration

SR OS supports the configuration modification of active measurement templates. That is, administratively disabling a measurement template that IP interfaces are actively referencing is not required. Modifying existing parameters causes interface delay tests that were referencing the modified template to terminate the current sample and aggregate measurement windows, and to start new measurement windows using the updated template parameters. The previous historical results are maintained, but the State field of the measurement window coinciding with the change indicates "Terminated". Changing the description or the **last-reported-delay-hold** configuration does not cause a termination of the current sample and aggregate measurement windows.

A measurement template cannot be removed if interfaces are referencing that template.

#### 4.1.1.5 Displaying link measurements

The following **show** commands are available to display the link measurement templates:

- The **show>test-oam>link-measurement measurement-template** reports the measurement templates and the number of total and active interface references.
- The **show>test-oam>link-measurement measurement-template-using** command reports the measurement template, associated router interfaces and the protocols.
- The **show>test-oam>link-measurement measurement-template *template-name*** command reports the specific measurement template and the associated configuration parameters, with the number of active interfaces and total references to this templates.

### 4.1.2 Interface assignment

The test criteria-specific link measurement configuration is under the link measurement template. Because the delay test is executed from the base router interface, a component of this configuration is required in the **config>router>interface>if-attribute> delay** context.

#### 4.1.2.1 IP addressing

To enable dynamic measurements for the interface, the user must configure a link measurement template and enable the test protocol using the **ipv4** or **ipv6** command. The link measurement template does not include interface-specific requirements, such as the IP protocol encapsulating the test packet or IP source and destination addressing. It is possible to enable the IPv4 or IPv6 protocol under the **dynamic>twamp-light** context without including any source or destination information.

When the IPv4 protocol is enabled with no addressing configured, the source address is automatically assigned to the primary IPv4 address of the IP interface. The destination address is automatically assigned if the primary IPv4 address has a prefix length of 30 or 31. In other cases, such as shorter prefix lengths or unnumbered interfaces, the destination address cannot be resolved and must be configured manually. The **source** and **destination** commands take precedence over the auto-assigned addressing; the IPv4 addresses must be unicast.

IPv4 auto-assigned addressing is not updated for operationally up interface delay tests when the IP addressing associated with that interface is changed. Nokia suggest the following options to update the auto-assigned addressing.

- Administratively disable and enable the protocol used for the interface delay test.
- Disable and enable the IP interface under which the IP address has changed.

When the IPv6 protocol is enabled without any source address, the system uses the link-local address associated with the interface as the source. If there is no destination address configuration, the destination discovery process is initiated if the associated measurement-template assigned to this interface has **configure test-oam link-measurement measurement-template twamp-light ipv6-destination-discovery** enabled.

The source and destination can be globally routable unicast addresses of the link identifying the directly-connected peers or the link local addresses connecting the peers. The link local address must follow the format fe80::/64 as described in RFC 4291, *IP Version 6 Addressing Architecture*.

TWAMP Light test packets consult the routing table to determine how to reach the destination. The test should be configured to use local IP interface source and directly connected IP peer interface destination addressing to ensure the packet egresses and returns over the same IP interface. The destination must be reachable from the IP interface where the interface delay test is configured. Using indirect IP addressing, such as unnumbered interfaces, does not guarantee that the measurement is reporting the delay for the expected interface.

Only one protocol, IPv4 or IPv6, can be enabled for an interface delay test at any time.

Interfaces defined as **loopback** do not support interface delay tests and are an invalid interface type. The configuration exists under these interfaces, but a detectable transmission error prevents the sending of packets.

The system interface does not support interface delay tests and the configuration is hidden.

#### 4.1.2.2 Test initialization

When the link measurement template is assigned to an IP interface, the audit process determines the operational state of the test. The interface delay test transitions to operationally up if the following conditions are met:

- The associated measurement template is administratively enabled.
- There is an administratively enable test protocol, configured using the **ipv4** or **ipv6** command.
- The system resources are available to start the test.

Further validation determines if there are any underlying conditions that are considered detectable transmission errors, which are listed in [Table 16: Detectable transmission errors](#) .

Table 16: Detectable transmission errors

Detectable Tx error	Description	Prevents transmission
None	No detectable errors	No
interfaceDown	The link measurement test is configured on an IP interface with an operationally down state	Yes
UnexpectedError	Router resources not available	No
noRoute	The routing lookup has failed to resolve the destination as reachable from the interface where the test is configured. The packet is transmitted out of the interface resolved by the routing engine.	No
sourceIpNotLocal	The source IP address configured for the test is not local to the system	No
invalidDestIp	The destination IP address is not valid. This may occur because of a configuration error or an attempt to use auto assignment in conditions that are not supported.	Yes
invalidInterfaceType	The link measurement test is configured under an interface that does not support these test types (such as loopback interfaces)	Yes
sameSourceIpDestIp	A configuration error that indicates the source and destination IP addresses are the same	Yes

When all audit conditions successfully pass, the delay collection begins. When no thresholds are configured, the test collects delay information as history, but without at least one configured threshold value, reporting updates to the routing engine are disabled. If at least one threshold is configured, the interface enters the first report phase. Because no previous delay value has been reported, the first measurement window with a configured threshold that completes with integrity triggers the delay measurement report. After this benchmark is set, all subsequent thresholds use the delay measurement last reported as the comparison.

### 4.1.2.3 History and results

Active interface delay tests retain 50 sample windows and 20 aggregate sample windows in history. The current measurement windows and historical results are not maintained across CPM switchovers. The delay measurement last reported is maintained after a CPM switchover to retain the baseline. The interface does not enter the first reporting phase following a CPM switchover.

Several states exist to indicate the state of the measurement window:

**InProgress**

The measurement window is open and collecting results.

**Completed**

This state indicates natural completion of the delay test.

**Terminated**

A configuration change to the measurement template being referenced, or an administrative action or system event caused the operational state to change to down, consequently preempting and causing an abnormal termination.

**SwReported**

The sample window has triggered a delay measurement last report.

**AswReported**

The aggregate sample window has triggered a delay measurement last report.

#### 4.1.2.4 Static versus dynamic

SR OS supports the configuration of **static** and **dynamic** delay measurement, and provides a configuration parameter to determine precedence. The dynamic delay measurements use the link measurement templates and interface delay test packets to gather results. The static delay is a configured value. Both static delay and dynamic delay can be configured under the same interface. Both are reported to the routing engine based on their own rules. The routing engine must be informed on how to interpret this condition and which metric to advertise. The delay selection, configured using the **delay-selection** command, provides four options to customize the handling. The configuration options allow for only one to be considered (using the **static** or **dynamic** options) or which is to be preferred should both exist (using the **static-preferred** or **dynamic-preferred** options). There may be an amount of time between the interface initialization and the reporting of a valid dynamic delay value. This case may require some deployments to configure a static delay value to fill the time gap while waiting for the link measurement first report. In this case, preferring dynamic allows the routing engine to advertise the static value until the dynamic delay report is made.

#### 4.1.2.5 Displaying interface delay tests

The user can use **show** commands to display the link measurement templates.

The **show test-oam link-measurement interface *interface-name* [detail | aggregate | sample debug-counters]** command reports the interface-specific delay test information, such as operational data, reporting information, the historical results for sample and aggregate sample windows, and a count of unexpected conditions for response packets.

The **show test-oam link-measurement interfaces** command reports a list of all interfaces that are configured with link-measurement. The report includes all the basic operational summaries such as the interface name, operational state, protocol, errors, reporting, delay metric of interest, last delay, and the timestamp of the last delay.

## 4.2 OAM performance monitoring

OAM Performance Monitoring (OAM-PM) provides an architecture for gathering and computing Key Performance Indicators (KPIs) using standard protocols and a robust collection model. The architecture comprises the following foundational components:

- **Session**

This is the overall collection of different tests, test parameters, measurement intervals, and mappings to configured storage models. It is the overall container that defines the attributes of the session.

- **Standard PM packets**

These are the protocols defined by various standards bodies, which contain the necessary fields to collect statistical data for the performance attribute they represent. OAM-PM leverages single-ended protocols. Single-ended protocols typically follow a message response model: message sent by a launch point, response updated, and reflected by a responder.

- **Measurement Intervals (MI)**

These are time-based non-overlapping windows that capture all results that are received in that window of time

- **Data structures**

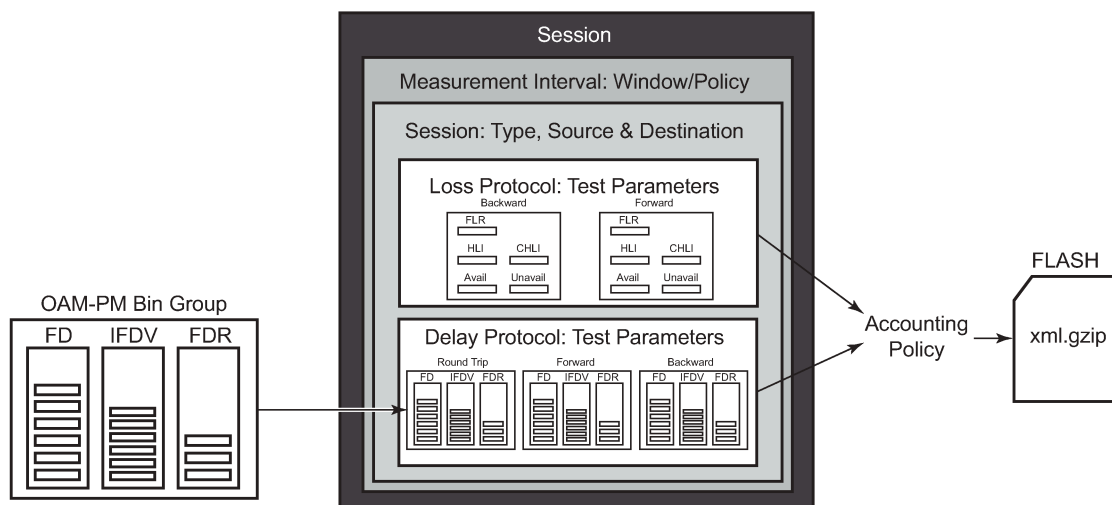
These are the unique counters and measurement results that represent the specific protocol

- **Bin group**

These are ranges in microseconds that count the results that fit into the range

**Figure 60: OAM-PM architecture hierarchy** shows the hierarchy of the architecture. This figure is only meant to show the relationship between the components. It is not meant to depict all details of the required parameters.

*Figure 60: OAM-PM architecture hierarchy*



al\_0386

OAM-PM configurations are not dynamic environments. All aspects of the architecture must be carefully considered before configuring the various architectural components, making external references to other related components, or activating the OAM-PM architecture. No modifications are allowed to any components that are active or have any active sub-components. Any function being referenced

by an active OAM-PM function or test cannot be modified or shut down. For example, to change any configuration element of a session, all active tests must be in a shutdown state. To change any bin group configuration (described later in this section), all sessions that reference the bin group must have every test shut down. The description parameter is the only exception to this rule.

Session source and destination configuration parameters are not validated by the test that uses that information. When the test is activated with a **no shutdown** command, the test engine attempts to send the test packets even if the session source and destination information does not accurately represent the entity that must exist to successfully transmit packets. If the entity does not exist, the transmit count for the test is zero.

OAM-PM is not a hitless operation. If a high availability event occurs that causes the backup CPM or CPIOM to become the active CPM or CPIOM, or when ISSU functions are performed, the test data is not correctly reported. There is no synchronization of state between the active and the backup control modules. All OAM-PM statistics stored in volatile memory are lost. When the reload or high availability event is completed and all services are operational, the OAM-PM functions commence.

It is possible that during times of network convergence, high CPU utilizations, or contention for resources, OAM-PM may not be able to detect changes to an egress connection or allocate the necessary resources to perform its tasks.

## 4.2.1 Session

The session is the overall collection of test information fields, including the test parameters, measurement intervals, and mapping to configured storage models. The container defines the attributes of the session. The fields are as follows:

- **session type**

The impetus of the test, which is either proactive (default) or on-demand. Individual test timing parameters are influenced by this setting. A proactive session starts immediately following the execution of a **no shutdown** command for the test. A proactive test continues to execute until a manual shutdown stops the individual test. All previous memory allocated to the test session is cleared when the new memory is allocated during the no shutdown. Any results not collected from volatile memory is permanently lost. On-demand tests also start immediately following the **no shutdown** command. However, the operator can override the **no test-duration** default and configure a fixed amount of time that the test executes, up to 24 hours (86 400 seconds).

If an on-demand test is configured with a test duration, it is important to shut down tests when they are completed. In the event of a high availability event causing the backup CPM or CPIOM to become the active CPM or CPIOM, all on-demand tests that have a test duration statement restart and run for the configured amount of time regardless of their progress on the previously active CPM or CPIOM.

- **test family**

The main branch of testing that addresses a specific technology. The available tests for the session are based on the test family. The destination, source, and priority are common to all tests under the session and are defined separately from the individual test parameters.

- **test parameters**

The parameters included in individual tests, as well as the associated parameters including start and stop times and the ability to activate and deactivate the individual test.

- **measurement interval**

The assignment of collection windows to the session with the appropriate configuration parameters and accounting policy for that specific session.

The session can be viewed as the single container that brings all aspects of individual tests and the various OAM-PM components together. If any aspects of the session are incomplete, the individual test cannot be activated with a **no shutdown** command, and an "Invalid Ethernet session parameters" error occurs.

## 4.2.2 Standard PM packets

A number of standards bodies define performance monitoring packets that can be sent from a source, processed, and responded to by a reflector. The protocols available to carry out the measurements are based on the test family type configured for the session.

Ethernet PM delay measurements are carried out using the Two-Way Delay Measurement Protocol version 1 (DMMv1) defined in Y.1731 by ITU-T. This allows for the collection of Frame Delay (FD), InterFrame Delay Variation (IFDV), Frame Delay Range (FDR), and Mean Frame Delay (MFD) measurements for round-trip, forward, and backward directions.

DMMv1 adds the following to the original DMM definition:

- the Flag Field (1 bit – LSB) is defined as the Type (Proactive=1 | On-Demand=0)
- the TestID TLV (32 bits) is carried in the Optional TLV portion of the PDU

DMMv1 and DMM are backwards compatible and the interaction is defined in Y.1731 ITU-T-2011 Section 11 "OAM PDU validation and versioning."

Ethernet PM loss measurements are carried out using Synthetic Loss Measurement (SLM), which is defined in Y.1731 by the ITU-T. This allows for the calculation of Frame Loss Ratio (FLR) and availability.

A session can be configured with one or more tests. Depending on the session test type family, one or more test configurations may need to be included in the session to gather both delay and loss performance information. Each test that is configured shares the common session parameters and the common measurement intervals. However, each test can be configured with unique per-test parameters. Using Ethernet as an example, both DMM and SLM would be required to capture both delay and loss performance data.

Each test must be configured with a TestID as part of the test parameters, which uniquely identifies the test within the specific protocol. A TestID must be unique within the same test protocol. Using Ethernet as an example, DMM and SLM tests within the same session can use the same TestID because they are different protocols. However, if a TestID is applied to a test protocol (like DMM or SLM) in any session, it cannot be used for the same protocol in any other session. When a TestID is carried in the protocol, as it is with DMM and SLM, this value does not have global significance. When a responding entity must index for the purpose of maintaining sequence numbers, as in the case of SLM, the TestID, source MAC, and destination MAC are used to maintain the uniqueness of the responder. This means that the TestID has only local, and not global, significance.

## 4.2.3 Measurement intervals

A measurement interval is a window of time that compartmentalizes the gathered measurements for an individual test that has occurred during that time. Allocation of measurement intervals, which equates to system memory, is based on the metrics being collected. This means that when both delay and loss metrics are being collected, they allocate their own set of measurement intervals. If the operator



is executing multiple delay and loss tests under a single session, multiple measurement intervals are allocated, with one interval allocated per criteria per test.

Measurement intervals can be 15 minutes (**15-min**), one hour (**1-hour**), and 1 day (**1-day**) in duration. The **boundary-type** defines the start of the measurement interval and can be aligned to the local time-of-day clock, with or without an optional offset. The **boundary-type** can be aligned using the **test-aligned** option, which means that the start of the measurement interval coincides with the activation of the test. By default, the start boundary is clock-aligned without an offset. When this configuration is deployed, the measurement interval starts at zero, in relation to the length.

When a boundary is clock-aligned and an offset is configured, the specified amount of time is applied to the measurement interval. Offsets are configured on a per-measurement interval basis and only applicable to clock-aligned measurement intervals. Only offsets less than the measurement interval duration are allowed. [Table 17: Measurement interval start times](#) lists examples of the start times of each measurement interval.

*Table 17: Measurement interval start times*

Offset	15-min	1-hour	1-day
0 (default)	00, 15, 30, 45	00 (top of the hour)	midnight
10 minutes	10, 25, 40, 55	10 min after the hour	10 min after midnight
30 minutes	rejected	30 min after the hour	30 min after midnight
60 minutes	rejected	rejected	01:00 AM

Although test-aligned approaches may seem beneficial for simplicity, there are some drawbacks that need to be considered. The goal of the time-based and well-defined collection windows allows for the comparison of measurements across common windows of time throughout the network and for relating different tests or sessions. It is suggested that proactive sessions use the default clock-aligned boundary type. On-demand sessions may use test-aligned boundaries. On-demand tests are typically used for troubleshooting or short term monitoring that does not require alignment or comparison to other PM data.

The statistical data collected and the computed results from each measurement interval are maintained in volatile system memory by default. The number of intervals stored is configurable per measurement interval. Different measurement intervals have different defaults and ranges. The **interval-stored** parameter defines the number of completed individual test runs to store in volatile memory. There is an additional allocation to account for the active measurement interval.

To look at the statistical information for the individual tests and a specific measurement interval stored in volatile memory, the **show oam-pm statistics ... interval-number** command can be used. If there is an active test, it can be viewed by using the interval number 1. In this case, the first completed record would be interval number 2, and previously completed records would increment up to the maximum intervals stored value plus one.

As new tests for the measurement interval are completed, the older entries are renumbered to maintain their relative position to the current test. If the retained test data for a measurement interval consumes the final entry, any subsequent entries cause the removal of the oldest data.

There are drawbacks to this storage model. Any high availability function that causes an active CPM or CPIOM switch flushes the results that are in volatile memory. Another consideration is the large amount of system memory consumed using this type of model. Considering the risks and resource consumption this model incurs, an alternate method of storage is supported. An accounting policy can be applied to each measurement interval to write the completed data in system memory to non-volatile flash memory

in an XML format. The amount of system memory consumed by historically completed test data must be balanced with an appropriate accounting policy.

Nokia recommends that only necessary data be stored in non-volatile memory to avoid unacceptable risk and unnecessary resource consumption. It is further suggested that a large overlap between the data written to flash memory and stored in volatile memory is unnecessary.

The statistical information in system memory is also available through SNMP. If this method is chosen, a balance must be struck between the intervals retained and the times at which the SNMP queries collect the data. Determining the collection times through SNMP must be done with caution. If a file is completed while another file is being retrieved through SNMP, the indexing changes to maintain the relative position to the current run. Correct spacing of the collection is key to ensuring data integrity.

[Table 18: OAM-PM XML keywords and MIB reference](#) describes the keywords and MIB references contained in the OAM-PM XML file.

Table 18: OAM-PM XML keywords and MIB reference

XML file keyword	Description	TIMETRA-OAM-PM-MIB object
oampm	—	None - header only
<b>Keywords shared by all OAM-PM protocols</b>		
sna	OAM-PM session name	tmnxOamPmCfgSessName
mi	Measurement Interval record	None - header only
dur	Measurement Interval duration (minutes)	tmnxOamPmCfgMeasIntvlDuration (enumerated)
ivl	measurement interval number	tmnxOamPmStsIntvlNum
sta	Start timestamp	tmnxOamPmStsBaseStartTime
ela	Elapsed time in seconds	tmnxOamPmStsBaseElapsedTime
ftx	Frames sent	tmnxOamPmStsBaseTestFramesTx
frx	Frames received	tmnxOamPmStsBaseTestFramesRx
sus	Suspect flag	tmnxOamPmStsBaseSuspect
<b>dmm</b>	<b>Delay Record</b>	None - header only
mdr	minimum frame delay, round-trip	tmnxOamPmStsDelayDmm2wyMin
xdr	maximum frame delay, round-trip	tmnxOamPmStsDelayDmm2wyMax
adr	average frame delay, round-trip	tmnxOamPmStsDelayDmm2wyAvg
mdf	minimum frame delay, forward	tmnxOamPmStsDelayDmmFwdMin
xdf	maximum frame delay, forward	tmnxOamPmStsDelayDmmFwdMax
adf	average frame delay, forward	tmnxOamPmStsDelayDmmFwdAvg

XML file keyword	Description	TIMETRA-OAM-PM-MIB object
mdb	minimum frame delay, backward	tmnxOamPmStsDelayDmmBwdMin
xdb	maximum frame delay, backward	tmnxOamPmStsDelayDmmBwdMax
adb	average frame delay, backward	tmnxOamPmStsDelayDmmBwdAvg
mvr	minimum inter-frame delay variation, round-trip	tmnxOamPmStsDelayDmm2wyMin
xvr	maximum inter-frame delay variation, round-trip	tmnxOamPmStsDelayDmm2wyMax
avr	average inter-frame delay variation, round-trip	tmnxOamPmStsDelayDmm2wyAvg
mvf	minimum inter-frame delay variation, forward	tmnxOamPmStsDelayDmmFwdMin
xvf	maximum inter-frame delay variation, forward	tmnxOamPmStsDelayDmmFwdMax
avf	average inter-frame delay variation, forward	tmnxOamPmStsDelayDmmFwdAvg
mvb	minimum inter-frame delay variation, backward	tmnxOamPmStsDelayDmmBwdMin
xvb	maximum inter-frame delay variation, backward	tmnxOamPmStsDelayDmmBwdMax
avb	average inter-frame delay variation, backward	tmnxOamPmStsDelayDmmBwdAvg
mrr	minimum frame delay range, round-trip	tmnxOamPmStsDelayDmm2wyMin
xrr	maximum frame delay range, round-trip	tmnxOamPmStsDelayDmm2wyMax
arr	average frame delay range, round-trip	tmnxOamPmStsDelayDmm2wyAvg
mrf	minimum frame delay range, forward	tmnxOamPmStsDelayDmmFwdMin
xrf	maximum frame delay range, forward	tmnxOamPmStsDelayDmmFwdMax
arf	average frame delay range, forward	tmnxOamPmStsDelayDmmFwdAvg
mrB	minimum frame delay range, backward	tmnxOamPmStsDelayDmmBwdMin
xrb	maximum frame delay range, backward	tmnxOamPmStsDelayDmmBwdMax
arb	average frame delay range, backward	tmnxOamPmStsDelayDmmBwdAvg
fdr	frame delay bin record, round-trip	None - header only
fdf	frame delay bin record, forward	None - header only
fdb	frame delay bin record, backward	None - header only

XML file keyword	Description	TIMETRA-OAM-PM-MIB object
fvr	inter-frame delay variation bin record, round-trip	None - header only
fvf	inter-frame delay variation bin record, forward	None - header only
fvb	inter-frame delay variation bin record, backward	None - header only
frr	frame delay range bin record, round-trip	None - header only
frf	frame delay range bin record, forward	None - header only
frb	frame delay range bin record, backward	None - header only
lbo	Configured lower bound of the bin	tmnxOamPmCfgBinLowerBound
cnt	Number of measurements within the configured delay range.  Note that the session_name, interval_duration, interval_number, {fd, fdr, ifdv}, bin_number, and {forward, backward, round-trip} indexes are all provided by the surrounding XML context.	tmnxOamPmStsDelayDmmBinFwdCount tmnxOamPmStsDelayDmmBinBwdCount tmnxOamPmStsDelayDmmBin2wyCount
<b>slm</b>	<b>Synthetic Loss Measurement Record</b>	None - header only
txf	Transmitted frames in the forward direction	tmnxOamPmStsLossSlmTxFwd
rxf	Received frames in the forward direction	tmnxOamPmStsLossSlmRxFwd
txb	Transmitted frames in the backward direction	tmnxOamPmStsLossSlmTxBwd
rxb	Received frames in the backward direction	tmnxOamPmStsLossSlmRxBwd
avf	Available count in the forward direction	tmnxOamPmStsLossSlmAvailIndFwd
avb	Available count in the backward direction	tmnxOamPmStsLossSlmAvailIndBwd
uvf	Unavailable count in the forward direction	tmnxOamPmStsLossSlmUnavIndFwd
uvb	Unavailable count in the backward direction	tmnxOamPmStsLossSlmUnavIndBwd
uaf	Undetermined available count in the forward direction	tmnxOamPmStsLossSlmUndtAviFwd
uab	Undetermined available count in the backward direction	tmnxOamPmStsLossSlmUndtAviBwd
uuf	Undetermined unavailable count in the forward direction	tmnxOamPmStsLossSlmUndtUnaviFwd

XML file keyword	Description	TIMETRA-OAM-PM-MIB object
uub	Undetermined unavailable count in the backward direction	tmnxOamPmStsLossSlmUndtUnaviBwd
hlf	Count of HLIs in the forward direction	tmnxOamPmStsLossSlmHliFwd
hlb	Count of HLIs in the backward direction	tmnxOamPmStsLossSlmHliBwd
chf	Count of CHLIs in the forward direction	tmnxOamPmStsLossSlmChliFwd
chb	Count of CHLIs in the backward direction	tmnxOamPmStsLossSlmChliBwd
mff	minimum FLR in the forward direction	tmnxOamPmStsLossSlmMinFlrFwd
xff	maximum FLR in the forward direction	tmnxOamPmStsLossSlmMaxFlrFwd
aff	average FLR in the forward direction	tmnxOamPmStsLossSlmAvgFlrFwd
mfb	minimum FLR in the backward direction	tmnxOamPmStsLossSlmMinFlrBwd
xfb	maximum FLR in the backward direction	tmnxOamPmStsLossSlmMaxFlrBwd
afb	average FLR in the backward direction	tmnxOamPmStsLossSlmAvgFlrBwd
<b>Imm</b>	<b>Frame loss measurement record</b>	None - header only
txf	Transmitted frames in the forward direction	tmnxOamPmStsLossLmmTxFwd
rxf	Received frames in the forward direction	tmnxOamPmStsLossLmmRxFwd
txb	Transmitted frames in the backward direction	tmnxOamPmStsLossLmmTxBwd
rxb	Received frames in the backward direction	tmnxOamPmStsLossLmmRxBwd
mff	minimum FLR in the forward direction	tmnxOamPmStsLossLmmMinFlrFwd
xff	maximum FLR in the forward direction	tmnxOamPmStsLossLmmMaxFlrFwd
aff	average FLR in the forward direction	tmnxOamPmStsLossLmmAvgFlrFwd
mfb	minimum FLR in the backward direction	tmnxOamPmStsLossLmmMinFlrBwd
xfb	maximum FLR in the backward direction	tmnxOamPmStsLossLmmMaxFlrBwd
afb	average FLR in the backward direction	tmnxOamPmStsLossLmmAvgFlrBwd
ave	Imm availability enabled/disabled	No TIMETRA-OAM-PM-MIB entry
avf	available count in the forward direction	tmnxOamPmStsLossLmmAvailIndFwd
avb	available count in the backward direction	tmnxOamPmStsLossLmmAvailIndBwd
uvf	unavailable count in the forward direction	tmnxOamPmStsLossLmmUnaviIndFwd

XML file keyword	Description	TIMETRA-OAM-PM-MIB object
uvb	unavailable count in the backward direction	tmnxOamPmStsLossLmmUnavlIndBwd
uaf	undetermined available count in the forward direction	tmnxOamPmStsLossLmmUndtAvlFwd
uab	undetermined available count in the backward direction	tmnxOamPmStsLossLmmUndtAvlBwd
uuf	undetermined unavailable count in the forward direction	tmnxOamPmStsLossLmmUndtUnavlFwd
uub	undetermined unavailable count in the backward direction	tmnxOamPmStsLossLmmUndtUnavlBwd
hlf	count of HLIs in the forward direction	tmnxOamPmStsLossLmmHliFwd
hlb	count of HLIs in the backward direction	tmnxOamPmStsLossLmmHliBwd
chf	count of CHLIs in the forward direction	tmnxOamPmStsLossLmmChliFwd
chb	count of CHLIs in the backward direction	tmnxOamPmStsLossLmmChliBwd
udf	undetermined delta-t in the forward direction	tmnxOamPmStsLossLmmUndetDelTsFwd
udb	undetermined delta-t in the backward direction	tmnxOamPmStsLossLmmUndetDelTsBwd
<b>TLD</b>	<b>TWAMP Light Delay Record</b>	None - header only
mdr	minimum frame delay, round-trip	tmnxOamPmStsDelayTwl2wyMin
xdr	maximum frame delay, round-trip	tmnxOamPmStsDelayTwl2wyMax
adr	average frame delay, round-trip	tmnxOamPmStsDelayTwl2wyAvg
mdf	minimum frame delay, forward	tmnxOamPmStsDelayTwlFwdMin
xdf	maximum frame delay, forward	tmnxOamPmStsDelayTwlFwdMax
adf	average frame delay, forward	tmnxOamPmStsDelayTwlFwdAvg
mdb	minimum frame delay, backward	tmnxOamPmStsDelayTwlBwdMin
xdb	maximum frame delay, backward	tmnxOamPmStsDelayTwlBwdMax
adb	average frame delay, backward	tmnxOamPmStsDelayTwlBwdAvg
mvr	minimum inter-frame delay variation, round-trip	tmnxOamPmStsDelayTwl2wyMin
xvr	maximum inter-frame delay variation, round-trip	tmnxOamPmStsDelayTwl2wyMax

XML file keyword	Description	TIMETRA-OAM-PM-MIB object
avr	average inter-frame delay variation, round-trip	tmnxOamPmStsDelayTwl2wyAvg
mvf	minimum inter-frame delay variation, forward	tmnxOamPmStsDelayTwlFwdMin
xvf	maximum inter-frame delay variation, forward	tmnxOamPmStsDelayTwlFwdMax
avf	average inter-frame delay variation, forward	tmnxOamPmStsDelayTwlFwdAvg
mvb	minimum inter-frame delay variation, backward	tmnxOamPmStsDelayTwlBwdMin
xvb	maximum inter-frame delay variation, backward	tmnxOamPmStsDelayTwlBwdMax
avb	average inter-frame delay variation, backward	tmnxOamPmStsDelayTwlBwdAvg
mrr	minimum frame delay range, round-trip	tmnxOamPmStsDelayTwl2wyMin
xrr	maximum frame delay range, round-trip	tmnxOamPmStsDelayTwl2wyMax
arr	average frame delay range, round-trip	tmnxOamPmStsDelayTwl2wyAvg
mrf	minimum frame delay range, forward	tmnxOamPmStsDelayTwlFwdMin
xrf	maximum frame delay range, forward	tmnxOamPmStsDelayTwlFwdMax
arf	average frame delay range, forward	tmnxOamPmStsDelayTwlFwdAvg
mrb	minimum frame delay range, backward	tmnxOamPmStsDelayTwlBwdMin
xrb	maximum frame delay range, backward	tmnxOamPmStsDelayTwlBwdMax
arb	average frame delay range, backward	tmnxOamPmStsDelayTwlBwdAvg
fdr	frame delay bin record, round-trip	None - header only
fdf	frame delay bin record, forward	None - header only
fdb	frame delay bin record, backward	None - header only
fvr	inter-frame delay variation bin record, round-trip	None - header only
fvf	inter-frame delay variation bin record, forward	None - header only
fvb	inter-frame delay variation bin record, backward	None - header only
frr	frame delay range bin record, round-trip	None - header only

XML file keyword	Description	TIMETRA-OAM-PM-MIB object
frf	frame delay range bin record, forward	None - header only
frb	frame delay range bin record, backward	None - header only
lbo	Configured lower bound of the bin	tmnxOamPmCfgBinLowerBound
cnt	Number of measurements within the configured delay range.  Note that the session_name, interval_duration, interval_number, {fd, fdr, ifdv}, bin_number, and {forward, backward, round-trip} indexes are all provided by the surrounding XML context.	tmnxOamPmStsDelayTwlBinFwdCount tmnxOamPmStsDelayTwlBinBwdCount tmnxOamPmStsDelayTwlBin2wyCount
<b>TLL</b>	<b>TWAMP Light Loss Record</b>	None - header only
txf	Transmitted frames in the forward direction	tmnxOamPmStsLossTwlTxFwd
rxf	Received frames in the forward direction	tmnxOamPmStsLossTwlRxFwd
txb	Transmitted frames in the backward direction	tmnxOamPmStsLossTwlTxBwd
rxb	Received frames in the backward direction	tmnxOamPmStsLossTwlRxBwd
avf	Available count in the forward direction	tmnxOamPmStsLossTwlAvailIndFwd
avb	Available count in the backward direction	tmnxOamPmStsLossTwlAvailIndBwd
uvf	Unavailable count in the forward direction	tmnxOamPmStsLossTwlUnavIndFwd
uvb	Unavailable count in the backward direction	tmnxOamPmStsLossTwlUnavIndBwd
uaf	Undetermined available count in the forward direction	tmnxOamPmStsLossTwlUndtAvlFwd
uab	Undetermined available count in the backward direction	tmnxOamPmStsLossTwlUndtAvlBwd
uuf	Undetermined unavailable count in the forward direction	tmnxOamPmStsLossTwlUndtUnavIFwd
uub	Undetermined unavailable count in the backward direction	tmnxOamPmStsLossTwlUndtUnavIBwd
hlf	Count of HLIs in the forward direction	tmnxOamPmStsLossTwlHliFwd
hlb	Count of HLIs in the backward direction	tmnxOamPmStsLossTwlHliBwd
chf	Count of CHLIs in the forward direction	tmnxOamPmStsLossTwlChliFwd
chb	Count of CHLIs in the backward direction	tmnxOamPmStsLossTwlChliBwd



XML file keyword	Description	TIMETRA-OAM-PM-MIB object
mff	minimum FLR in the forward direction	tmnxOamPmStsLossTwlMinFlrFwd
xff	maximum FLR in the forward direction	tmnxOamPmStsLossTwlMaxFlrFwd
aff	average FLR in the forward direction	tmnxOamPmStsLossTwlAvgFlrFwd
mfb	minimum FLR in the backward direction	tmnxOamPmStsLossTwlMinFlrBwd
xfb	maximum FLR in the backward direction	tmnxOamPmStsLossTwlMaxFlrBwd
afb	average FLR in the backward direction	tmnxOamPmStsLossTwlAvgFlrBwd
<b>dm</b>	<b>MPLS Delay Record</b>	<b>None - header only</b>
mdr	minimum frame delay, round-trip	tmnxOamPmStsDelayMpls2wyMin
xdr	maximum frame delay, round-trip	tmnxOamPmStsDelayMpls2wyMax
adr	average frame delay, round-trip	tmnxOamPmStsDelayMpls2wyAvg
mdf	minimum frame delay, forward	tmnxOamPmStsDelayMplsFwdMin
xdf	maximum frame delay, forward	tmnxOamPmStsDelayMplsFwdMax
adf	average frame delay, forward	tmnxOamPmStsDelayMplsFwdAvg
mdb	minimum frame delay, backward	tmnxOamPmStsDelayMplsBwdMin
xdb	maximum frame delay, backward	tmnxOamPmStsDelayMplsBwdMax
adb	average frame delay, backward	tmnxOamPmStsDelayMplsBwdAvg
mvr	minimum inter-frame delay variation, roundtrip	tmnxOamPmStsDelayMpls2wyMin
xvr	maximum inter-frame delay variation, roundtrip	tmnxOamPmStsDelayMpls2wyMax
avr	average inter-frame delay variation, roundtrip	tmnxOamPmStsDelayMpls2wyAvg
mvf	minimum inter-frame delay variation, forward	tmnxOamPmStsDelayMplsFwdMin
xvf	maximum inter-frame delay variation, forward	tmnxOamPmStsDelayMplsFwdMax
avf	average inter-frame delay variation, forward	tmnxOamPmStsDelayMplsFwdAvg
mvb	minimum inter-frame delay variation, backward	tmnxOamPmStsDelayMplsBwdMin
xvb	maximum inter-frame delay variation, backward	tmnxOamPmStsDelayMplsBwdMax

XML file keyword	Description	TIMETRA-OAM-PM-MIB object
avb	average inter-frame delay variation, backward	tmnxOamPmStsDelayMplsBwdAvg
mrr	minimum frame delay range, round-trip	tmnxOamPmStsDelayMpls2wyMin
xrr	maximum frame delay range, round-trip	tmnxOamPmStsDelayMpls2wyMax
arr	average frame delay range, round-trip	tmnxOamPmStsDelayMpls2wyAvg
mrf	minimum frame delay range, forward	tmnxOamPmStsDelayMplsFwdMin
xrf	maximum frame delay range, forward	tmnxOamPmStsDelayMplsFwdMax
arf	average frame delay range, forward	tmnxOamPmStsDelayMplsFwdAvg
mrb	minimum frame delay range, backward	tmnxOamPmStsDelayMplsBwdMin
xrb	maximum frame delay range, backward	tmnxOamPmStsDelayMplsBwdMax
arb	average frame delay range, backward	tmnxOamPmStsDelayMplsBwdAvg
fdr	frame delay bin record, round-trip	None - header only
fdf	frame delay bin record, forward	None - header only
fdb	frame delay bin record, backward	None - header only
fvr	inter-frame delay variation bin record, roundtrip	None - header only
fvf	inter-frame delay variation bin record, forward	None - header only
fvb	inter-frame delay variation bin record, backward	None - header only
frf	frame delay range bin record, round-trip	None - header only
frf	frame delay range bin record, forward	None - header only
frb	frame delay range bin record, backward	None - header only
cnt	The number of measurements within the configured delay range. Note that the session_name, interval_duration, interval_number, {fd, fdr, ifdv}, bin_number, and {forward, backward, round-trip} indices are all provided by the surrounding XML context.	tmnxOamPmStsDelayMplsBinFwdCount tmnxOamPmStsDelayMplsBinBwdCount tmnxOamPmStsDelayMplsBin2wyCount

By default, the 15-min measurement interval stores 33 test runs (32+1) with a configurable range of 1 to 96, and the 1-hour measurement interval stores 9 test runs (8+1) with a configurable range of 1 to 24. The

only storage for the 1-day measurement interval is 2 (1+1). This value for the 1-day measurement interval cannot be changed.

All three measurement intervals may be added to a single session if required. Each measurement interval that is included in a session is updated simultaneously for each test that is executing. If a measurement interval length is not required, it should not be configured.

In addition to the three predetermined length measurement intervals, a fourth "always on" raw measurement interval is allocated at test creation. Data collection for the raw measurement interval commences immediately following the execution of a **no shutdown** command. It is a valuable tool for assisting in real-time troubleshooting as it maintains the same performance information and relates to the same bins as the fixed length collection windows. The operator may clear the contents of the raw measurement interval and flush stale statistical data to look at current conditions. This measurement interval has no configuration options, cannot be written to flash memory, and cannot be disabled; it is a single never-ending window.

Memory allocation for the measurement intervals is performed when the test is configured. Volatile memory is not flushed until the test is deleted from the configuration; a high availability event causes the backup CPM or CPIOM to become the newly active CPM or CPIOM, or some other event clears the active CPM or CPIOM system memory. Shutting down a test does not release the allocated memory for the test.

Measurement intervals also include a suspect flag. The suspect flag is used to indicate that data collected in the measurement interval may not be representative. The flag is set to true only under the following conditions:

- The time-of-day clock is adjusted by more than 10 seconds.
- The test start does not align with the start boundary of the measurement interval. This would be common for the first execution for clock-aligned tests.
- The test is stopped before the end of the measurement interval boundary.

The suspect flag is not set when there are times of service disruption, maintenance windows, discontinuity, low packet counts, or other such events. Higher-level systems would be required to interpret and correlate those types of events for measurement intervals that executed during the time that relates to the specific interruption or condition. Because each measurement interval contains a start and stop time, the information is readily available for higher-level systems to discount the specific windows of time.

## 4.2.4 Data structures and storage

There are two main metrics that are the focus of OAM-PM: delay and loss. The different metrics have two unique storage structures and allocate their own measurement intervals for these structures. This occurs regardless of whether the performance data is gathered with a single packet or multiple packet types.

Unidirectional and round-trip results are stored for each delay metric. The delay metrics are as follows:

<b>Frame Delay</b>	the amount of time required to send and receive the packet
<b>InterFrame Delay Variation</b>	the difference in the delay metrics between two adjacent packets
<b>Frame Delay Range</b>	the difference between the minimum frame delay and the individual packet
<b>Mean Frame Delay</b>	the mathematical average for the frame delay over the entire window

FD, IFDV, and FDR statistics are binnable results. FD, IFDV, FDR, and MFD all include minimum, maximum, and average values. Unidirectional and round-trip results are stored for each metric.

Unidirectional frame delay and frame delay range measurements require exceptional time-of-day clock synchronization. If the time-of-day clock does not exhibit extremely tight synchronization, unidirectional measurements is not representative. In one direction, the measurement is artificially increased by the difference in the clocks. In the other direction, the measurement is artificially decreased by the difference in the clocks. This level of clocking accuracy is not available with NTP. To achieve this level of time-of-day clock synchronization, Precision Time Protocol (PTP) 1588v2 should be considered.

Round-trip metrics do not require clock synchronization between peers, because the four timestamps allow for accurate representation of the round-trip delay. The mathematical computation removes remote processing and any difference in time-of-day clocking. Round-trip measurements do require stable local time-of-day clocks.

Any delay metric that is negative is treated as zero and placed in bin 0, the lowest bin, which has a lower boundary of 0 microseconds.

Delay results are mapped to the measurement interval that is active when the result arrives back at the source.

There are no supported log events based on delay metrics.

Loss metrics are only unidirectional and report Frame Loss Ratio (FLR) and availability information. FLR is the computation of loss (lost/sent) over time. Loss measurements during periods of unavailability are not included in the FLR calculation as they are counted against the unavailability metric.

Availability requires relating three different functions. First, the individual probes are marked as available or unavailable based on sequence numbers in the protocol. A number of probes are rolled up into a small measurement window, typically 1 s. FLR is computed over all the probes in a small window. If the resulting percentage is higher than the configured threshold, the small window is marked as unavailable. If the resulting percentage is lower than the threshold, the small window is marked as available. A sliding window is defined as some number of small windows, typically 10. The sliding window is used to determine availability and unavailability events. Switching from one state to the other requires every small window in the sliding window to be the same state and different from the current state.

Availability and unavailability counters are incremented based on the number of small windows that have occurred in all available and unavailable windows.

Availability and unavailability using synthetic loss measurements is meant to capture the loss behavior for the service. It is not meant to capture and report on service outages or communication failures. Communication failures of a bidirectional or unidirectional nature must be captured using some other means of connectivity verification, alarming, or continuity checking. During times of complete or extended failure periods it becomes necessary to timeout individual test probes. It is not possible to determine the direction of the loss because no response packets are being received back on the source. In this case, the statistics calculation engine maintains the previous state, updating the appropriate directional availability or unavailability counter. At the same time, an additional per-direction undetermined counter is updated. This undetermined counter is used to indicate that the availability or unavailability statistics could not be determined for a number of small windows.

During connectivity outages, the higher-level systems can be used to discount the loss measurement interval, which covers the same span as the outage.

Availability and unavailability computations may delay the completion of a measurement interval. The declaration of a state change or the delay to a closing a measurement interval could be equal to the length of the sliding window and the timeout of the last packet. Closing of a measurement interval cannot occur until the sliding window has determined availability or unavailability. If the availability state is changing, and the determination is crossing two measurement intervals, the measurement interval does not complete until the declaration has occurred. Typically, standard bodies indicate the timeout per packet. In the case of Ethernet, DMMv1, and SLM, timeout values are set at 5 s and cannot be configured.

There are no log events based on availability or unavailability state changes.

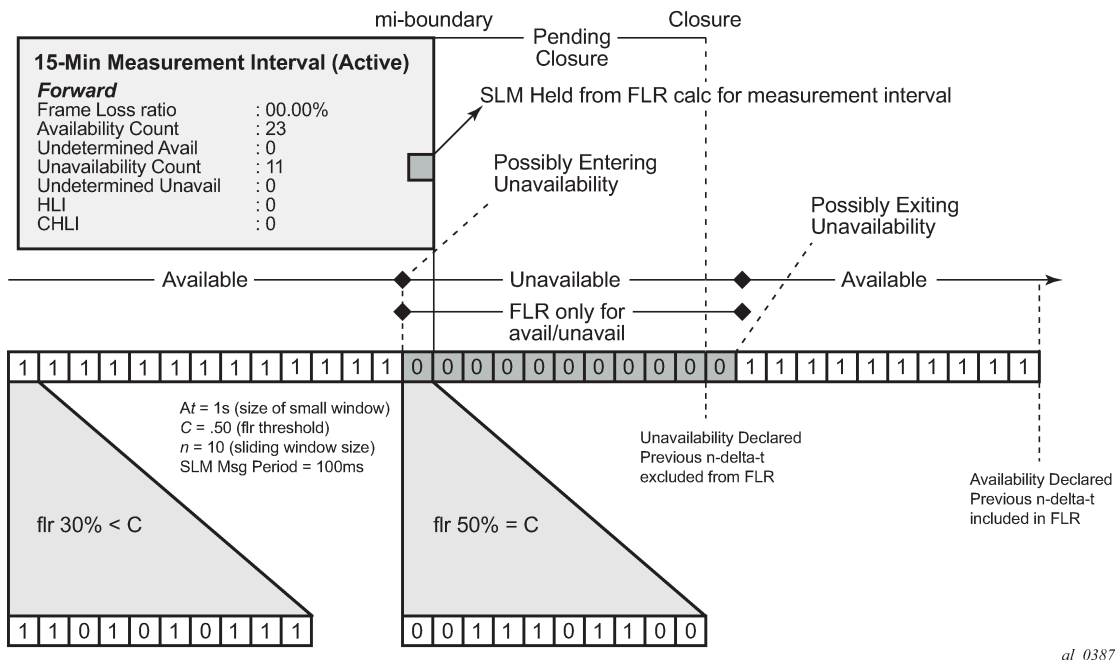
During times of availability, there can be times of high loss intervals (HLI) or consecutive high loss intervals (CHLI). These are indicators that the service was available but individual small windows or consecutive small windows experienced FLRs exceeding the configured acceptable limit. A HLI is any single small window that exceeds the configured FLR. This could equate to a severely errored second, assuming the small window is one second in length. A CHIL is a consecutive high loss interval that exceeds a consecutive threshold within the sliding window. Only one HLI is counted for a window.

Availability can only be reasonably determined with synthetic packets. This is because the synthetic packet is the packet being counted and provides a uniform packet flow that can be used for the computation. Transmit and receive counter-based approaches cannot reliably be used to determine availability because there is no guarantee that service data is on the wire, or the service data on the wire uniformity could make it difficult to make a declaration valid.

Figure 61: Evaluating and computing loss and availability shows loss in a single direction using synthetic packets, and demonstrates what happens when a possible unavailability event crosses a measurement interval boundary. In the diagram, the first 13 small windows are all marked available (1), which means that the loss probes that fit into each of those small windows did not equal or exceed a frame loss ratio of 50%. The next 11 small windows are marked as unavailable, which means that the loss probes that fit into each of those small windows were equal to or above a frame loss ratio of 50%. After the 10th consecutive small window of unavailability, the state transitions from available to unavailable. The 25th small window is the start of the new available state which is declared following the 10th consecutive available small window.

The frame loss ratio is 00.00%; this is because all the small windows that are marked as unavailable are counted toward unavailability, and are therefore excluded from impacting the FLR. If there were any small windows of unavailability that were outside of an unavailability event, they would be marked as HLI or CHLI and be counted as part of the frame loss ratio.

Figure 61: Evaluating and computing loss and availability



## 4.2.5 Bin groups

Bin groups are templates that are referenced by the session. Three types of binnable delay metric types are available: FD, IFDV, and FDR; all of which are available in forward, backward, and round-trip directions. Each of these metrics can have up to ten bin groups configured to group the results. Bin groups are configured by indicating a lower boundary. Bin 0 has a lower boundary that is always zero and is not configurable. The microsecond range of the bins is the difference between the adjacent lower boundaries. For example, **bin-type fd bin 1** configured with **lower-bound 1000** means that bin 0 captures all frame delay statistics results between 0 and 1 ms. Bin 1 captures all results above 1 ms and below the bin 2 lower boundary. The last bin configured represents the bin that collects all the results at and above that value. Not all ten bins have to be configured.

Each binnable delay metric type requires their own values for the bin groups. Each bin in a type is configurable for one value. It is not possible to configure a bin with different values for round-trip, forward, and backward. Consider the configuration of the boundaries that represent the important statistics for that specific service.

As stated earlier in this section, this is not a dynamic environment. If a bin group is being referenced by any active test, the bin group cannot shut down. To modify the bin group, it must be shut down. If the configuration of a bin group must be changed, and a large number of sessions are referencing the bin group, migrating existing sessions to a new bin group with the new parameters can be considered to reduce the maintenance window. To modify any session parameter, every test in the session must be shut down.

Bin group 1 is the default bin group. Every session requires a bin group to be assigned. By default, bin group 1 is assigned to every OAM-PM session that does not have a bin group explicitly configured. Bin group 1 cannot be modified. The bin group 1 configuration parameters are as follows:

```

-----
Configured Lower Bounds for Delay Measurement (DMM) Tests, in microseconds
-----
Group Description                Admin Bin  FD(us)  FDR(us)  IFDV(us)
-----
1  OAM PM default bin group (not*  Up    0        0         0         0
                               1      5000    5000     5000
                               2     10000    -         -
-----

```

## 4.2.6 Delay results streaming

Service Level Agreements (SLAs) typically require that performance data is collected over five-minute (or longer) measurement intervals. Network optimization tools require average performance values to be computed over shorter periods of time. The OAM-PM streaming function takes advantage of the OAM-PM architecture and test definitions to provide the basis for short window average results streaming.

The **delay-template** configuration allows users to define the common parameters: metric type, direction, length of the sample window, and integrity value. After the template is defined, it can be assigned to the appropriate technology to support tests for collection, calculation, and reporting. The results of the process are sent using on-change update notifications to subscribers.

The streaming function is supported for Ethernet DMM tests and IP TWAMP-light (delay) tests.

The updates are sent only if a subscription is registered for the on-change values. The keys are common for each individual test: **session** *session-name*, **metric** *metric-id*, and **newest-closed** *direction*. The following values are sent for each completed sample window:



### 4.2.8.1 Accounting policy configuration

```
config>log# info
-----
file-id 1
description "OAM PM XML file Paramaters"
location cf2:
rollover 10 retention 2
exit
accounting-policy 1
description "Default OAM PM Collection Policy for 15-min Bins"
record complete-pm
collection-interval 5
to file 1
no shutdown
exit
log-id 1
exit
-----
```

### 4.2.8.2 ETH-CFM configuration

```
config>eth-cfm# info
-----
domain 12 format none level 2
association 4 format string name "vpls4-0000001"
bridge-identifier 4
exit
ccm-interval 1
remote-mepid 30
exit
exit
```

### 4.2.8.3 Service configuration

```
config>service>vpls# info
-----
description "OAM PM Test Service to v30"
stp
shutdown
exit
sap 1/1/10:4.* create
eth-cfm
mep 28 domain 12 association 4 direction up
ccm-enable
mac-address 00:00:00:00:00:28
no shutdown
exit
exit
exit
sap 1/2/1:4.* create
exit
no shutdown
```



#### 4.2.8.4 OAM-PM configuration

```
config>oam-pm#info detail
-----
bin-group 2 fd-bin-count 10 fdr-bin-count 2 ifdv-bin-count 10 create
no description
bin-type fd
  bin 1
    lower-bound 1000
  exit
  bin 2
    lower-bound 2000
  exit
  bin 3
    lower-bound 3000
  exit
  bin 4
    lower-bound 4000
  exit
  bin 5
    lower-bound 5000
  exit
  bin 6
    lower-bound 6000
  exit
  bin 7
    lower-bound 7000
  exit
  bin 8
    lower-bound 8000
  exit
  bin 9
    lower-bound 10000
  exit
exit
bin-type fdr
  bin 1
    lower-bound 5000
  exit
exit
bin-type ifdv
  bin 1
    lower-bound 100
  exit
  bin 2
    lower-bound 200
  exit
  bin 3
    lower-bound 300
  exit
  bin 4
    lower-bound 400
  exit
  bin 5
    lower-bound 500
  exit
  bin 6
    lower-bound 600
  exit
  bin 7
    lower-bound 700
  exit
  bin 8
```

```

        lower-bound 800
    exit
    bin 9
        lower-bound 1000
    exit
    exit
    no shutdown
    exit
    session "eth-pm-service-4" test-family Ethernet session-
type proactive create
    bin-group 2
    no description
    meas-interval 15-mins create
        no accounting-policy
        boundary-type clock-aligned
        clock-offset 0
        intervals-stored 32
    exit
    Ethernet
        dest-mac 00:00:00:00:00:30
        priority 0
        source mep 28 domain 12 association 4
        dmm test-id 10004 create
            data-tlv-size 1000
            interval 1000
            no test-duration
            no shutdown
        exit
        slm test-id 10004 create
            data-tlv-size 1000
            flr-threshold 50
            no test-duration
            timing frames-per-delta-t 10 consec-delta-t 10 interval 100
                chli-threshold 4
            no shutdown
        exit
    exit
    exit
    exit

```

#### 4.2.8.5 Show and monitor commands

```

show oam-pm bin-group
-----
Configured Lower Bounds for Delay Measurement (DMM) Tests, in microseconds
-----
Group Description                               Admin Bin   FD(us)    FDR(us)   IFDV(us)
-----
1   OAM PM default bin group (not*             Up    0         0         0         0
    1         5000         5000         5000
    2         10000        -          -
-----
2                                       Up    0         0         0         0
    1         1000         5000         100
    2         2000         -          200
    3         3000         -          300
    4         4000         -          400
    5         5000         -          500
    6         6000         -          600
    7         7000         -          700
    8         8000         -          800
    9         10000        -          1000

```

```

-----
* indicates that the corresponding row element may have been truncated.

show oam-pm bin-group 2
-----
Configured Lower Bounds for Delay Measurement (DMM) Tests, in microseconds
-----
Group Description          Admin Bin    FD(us)    FDR(us)    IFDV(us)
-----
2                          Up    0         0         0         0
                          1     1000     5000     100
                          2     2000     -        200
                          3     3000     -        300
                          4     4000     -        400
                          5     5000     -        500
                          6     6000     -        600
                          7     7000     -        700
                          8     8000     -        800
                          9    10000     -       1000
-----

show oam-pm bin-group-using
=====
OAM Performance Monitoring Bin Group Configuration for Sessions
=====
Bin Group    Admin    Session                Session State
-----
2            Up      eth-pm-service-4      Act
-----
=====

show oam-pm bin-group-using bin-group 2
=====
OAM Performance Monitoring Bin Group Configuration for Sessions
=====
Bin Group    Admin    Session                Session State
-----
2            Up      eth-pm-service-4      Act
-----
=====

show oam-pm sessions test-family Ethernet
=====
OAM Performance Monitoring Session Summary for the Ethernet Test Family
=====
Session                State    Bin Group    Sess Type    Test Types
-----
eth-pm-service-4      Act      2            proactive    DMM SLM
=====

show oam-pm session "eth-pm-service-4" all
-----
Basic Session Configuration
-----
Session Name      : eth-pm-service-4
Description       : (Not Specified)
Test Family       : Ethernet          Session Type      : proactive
Bin Group         : 2
-----

-----
Ethernet Configuration
-----

```

```

Source MEP      : 28          Priority      : 0
Source Domain   : 12          Dest MAC Address : 00:00:00:00:00:30
Source Assoc'n  : 4
-----

```

```

-----
DMM Test Configuration and Status
-----

```

```

Test ID          : 10004          Admin State     : Up
Oper State       : Up             Data TLV Size   : 1000 octets
On-Demand Duration: Not Applicable On-Demand Remaining: Not Applicable
Interval         : 1000 ms
-----

```

```

-----
SLM Test Configuration and Status
-----

```

```

Test ID          : 10004          Admin State     : Up
Oper State       : Up             Data TLV Size   : 1000 octets
On-Demand Duration: Not Applicable On-Demand Remaining: Not Applicable
Interval         : 100 ms
CHLI Threshold   : 4 HLIs        Frames Per Delta-T : 10 SLM frames
Consec Delta-Ts  : 10            FLR Threshold    : 50%
-----

```

```

-----
15-mins Measurement Interval Configuration
-----

```

```

Duration         : 15-mins        Intervals Stored : 32
Boundary Type    : clock-aligned   Clock Offset      : 0 seconds
Accounting Policy : none
-----

```

```

-----
Configured Lower Bounds for Delay Measurement (DMM) Tests, in microseconds
-----

```

Group Description	Admin	Bin	FD(us)	FDR(us)	IFDV(us)
2	Up	0	0	0	0
		1	1000	5000	100
		2	2000	-	200
		3	3000	-	300
		4	4000	-	400
		5	5000	-	500
		6	6000	-	600
		7	7000	-	700
		8	8000	-	800
		9	10000	-	1000

```

-----

```

```

show oam-pm statistics session "eth-pm-service-4" dmm meas-interval 15-
mins interval-number 2 all
-----

```

```

Start (UTC)      : 2014/02/01 10:00:00      Status          : completed
Elapsed (seconds) : 900                      Suspect         : no
Frames Sent      : 900                      Frames Received : 900
-----

```

```

-----
Bin Type   Direction   Minimum (us)  Maximum (us)  Average (us)
-----
FD         Forward     0             8330          712
FD         Backward   143          11710         2605
FD         Round Trip 1118         14902         3111
-----

```

```

FDR      Forward      0      8330      712
FDR      Backward     143     11710     2605
FDR      Round Trip   0      13784     1990
IFDV     Forward      0      8330      431
IFDV     Backward     1      10436     800
IFDV     Round Trip   2      13542     1051
-----

```

```

-----
Frame Delay (FD) Bin Counts
-----

```

Bin	Lower Bound	Forward	Backward	Round Trip
0	0 us	624	53	0
1	1000 us	229	266	135
2	2000 us	29	290	367
3	3000 us	4	195	246
4	4000 us	7	71	94
5	5000 us	5	12	28
6	6000 us	1	7	17
7	7000 us	0	1	5
8	8000 us	1	4	3
9	10000 us	0	1	5

```

-----
Frame Delay Range (FDR) Bin Counts
-----

```

Bin	Lower Bound	Forward	Backward	Round Trip
0	0 us	893	875	873
1	5000 us	7	25	27

```

-----
Inter-Frame Delay Variation (IFDV) Bin Counts
-----

```

Bin	Lower Bound	Forward	Backward	Round Trip
0	0 us	411	162	96
1	100 us	113	115	108
2	200 us	67	84	67
3	300 us	56	67	65
4	400 us	36	46	53
5	500 us	25	59	54
6	600 us	25	27	38
7	700 us	29	34	22
8	800 us	41	47	72
9	1000 us	97	259	325

```

show oam-pm statistics session "eth-pm-service-4" slm meas-interval 15-
mins interval-number 2
-----

```

```

Start (UTC)      : 2014/02/01 10:00:00      Status           : completed
Elapsed (seconds) : 900                    Suspect          : no
Frames Sent      : 9000                    Frames Received  : 9000
-----

```

```

-----
Frames Sent      Frames Received
-----
Forward          9000                    9000
Backward         9000                    9000
-----

```

```
-----
Frame Loss Ratios
-----
```

	Minimum	Maximum	Average
Forward	0.000%	0.000%	0.000%
Backward	0.000%	0.000%	0.000%

```
-----
Availability Counters (Und = Undetermined)
-----
```

	Available	Und-Avail	Unavailable	Und-Unavail	HLI	CHLI
Forward	900	0	0	0	0	0
Backward	900	0	0	0	0	0

```
show oam-pm statistics session "eth-pm-service-4" dmm meas-interval raw
```

```
-----
Start (UTC)       : 2014/02/01 09:43:58      Status          : in-progress
Elapsed (seconds) : 2011                    Suspect         : yes
Frames Sent       : 2011                    Frames Received : 2011
-----
```

```
-----
```

Bin Type	Direction	Minimum (us)	Maximum (us)	Average (us)
FD	Forward	0	11670	632
FD	Backward	0	11710	2354
FD	Round Trip	1118	14902	2704
FDR	Forward	0	11670	611
FDR	Backward	0	11710	2353
FDR	Round Trip	0	13784	1543
IFDV	Forward	0	10027	410
IFDV	Backward	0	10436	784
IFDV	Round Trip	0	13542	1070

```
-----
```

```
-----
Frame Delay (FD) Bin Counts
-----
```

Bin	Lower Bound	Forward	Backward	Round Trip
0	0 us	1465	252	0
1	1000 us	454	628	657
2	2000 us	62	593	713
3	3000 us	8	375	402
4	4000 us	11	114	153
5	5000 us	7	26	41
6	6000 us	2	10	20
7	7000 us	0	2	8
8	8000 us	1	10	11
9	10000 us	1	1	6

```
-----
```

```
-----
Frame Delay Range (FDR) Bin Counts
-----
```

Bin	Lower Bound	Forward	Backward	Round Trip
0	0 us	2001	1963	1971

```

1          5000 us          11          49          41
-----
-----
Inter-Frame Delay Variation (IFDV) Bin Counts
-----
Bin      Lower Bound      Forward      Backward      Round Trip
-----
0         0 us           954         429         197
1        100 us          196         246         197
2        200 us          138         168         145
3        300 us          115         172         154
4        400 us           89          96         136
5        500 us           63          91         108
6        600 us           64          53          89
7        700 us           61          55          63
8        800 us          112          82         151
9       1000 us          219         619         771
-----

show oam-pm statistics session "eth-pm-service-4" slm meas-interval raw
-----
Start (UTC)      : 2014/02/01 09:44:03      Status      : in-progress
Elapsed (seconds) : 2047                  Suspect     : yes
Frames Sent      : 20470                  Frames Received : 20469
-----

-----
Frames Sent      Frames Received
-----
Forward          20329          20329
Backward         20329          20329
-----

-----
Frame Loss Ratios
-----
Minimum      Maximum      Average
-----
Forward      0.000%      0.000%      0.000%
Backward     0.000%      0.000%      0.000%
-----

-----
Availability Counters (Und = Undetermined)
-----
Available      Und-Avail      Unavailable      Und-Unavail      HLI      CHLI
-----
Forward        2033           0                0                0                0        0
Backward       2033           0                0                0                0        0
-----

```

The **monitor** command can be used to automatically update the statistics for the raw measurement interval.

### 4.3 Service Assurance Agent

The Service Application Agent (SAA) tool allows operators to configure several tests that provide performance information such as delay, jitter, and loss of services or network segments. The test results

are saved in SNMP tables or summarized XML files. These results can be collected and reported on using network management systems.

SAA uses resources allocated to various OAM processes. These processes are not dedicated to SAA but are shared throughout the system. [Table 19: SAA test and descriptions](#) describes the logical groups of different OAM functions.

*Table 19: SAA test and descriptions*

Test	Description
Background	Tasks are configured outside the SAA hierarchy that consume OAM task resources. Specifically, these include SDP keepalive, static route CPE check, filter redirect policy, ping test, and VRRP policy host unreachable. These are critical tasks that ensure the network operation and may affect data forwarding or network convergence.
SAA Continuous	SAA tests configured as continuous (always scheduled)
SAA non-continuous	SAA tests that are not configured as continuous, and are scheduled outside the SAA application. The <b>oam saa test-name start</b> command is required to initiate the test run.
Non-SAA (Directed)	Tasks that do not include any configuration under SAA. These tests are SNMP or via the CLI that is used to troubleshoot or profile network condition. This would take the form "oam test-type", or ping or traceroute with the specific test parameters.

Y.1731 defines two approaches for measuring frame delay and frame delay variation: single-ended and dual-ended. SAA supports the single-ended approach.

SAA test types are restricted to tests that use a request response mechanism; that is, single-ended tests. Dual-ended tests that initiate the test on one node but require the statistical gathering on the other node are not supported under SAA.

Post-processing analysis of individual test runs can be used to determine the success or failure of these runs. The operator can set rising and lowering thresholds for delay, jitter, and loss. Exceeding the threshold causes the Last Test Result field to display the Failed keyword. A trap can be generated when the test fails. The operator can also configure a probe failure threshold and trap when these thresholds are exceeded.

Each supported test type has test-specific configuration properties. Not all options, intervals, and parameters are available for all tests. Some configuration parameters, such as the sub-second probe interval, require specific hardware.

Trace type tests apply the timeout to each individual packet, which may affect spacing. Packet timeout is required to move from one probe to the next probe. For tests that do not require this type of behavior, typically ping and ETH-CFM PM functions, the probes are sent at the specified interval and the timeout is only applied at the end of the test if any probe is lost during the run. When the timeout is applied at the end of the run, the test is considered complete either when all responses are received or the timeout expires at the end of the test run. For tests marked continuous (always scheduled), the spacing between runs may be delayed by the timeout value when a packet is lost. The test run is complete when all probes have either been received back or the timeout value has expired.

To preserve system resources, specifically memory, the operator should only store summarized history results. By default, summary results are stored for tests configured with sub-second probe intervals



or a probe count above 100, or is written to a file. By default, per-probe information is stored for tests configured with an interval of one second counters or above and probe counts of 100 or less, and is not written to a file. The operator may choose to override these defaults using the **probe-history {keep | drop | auto}** command options. The **auto** option sets the preceding defaults. The other options override the default retention schemes based on the operator requirements. The **keep** option retains and stores per-probe information and the **drop** option stores summary-only information. The probe data can be viewed using the **show saa test** command. If the per-probe information is retained, this data is available at the completion of the test run. The summary data is updated throughout the test run. The overall memory system usage is available using the **show system memory-pools** command. The OAM entry represents the overall memory usage. This includes the history data stored for SAA tests. A **clear saa testname** option is available to release the memory and flush test results.

SAA-launched tests maintain the two most recently completed tests and one in progress test. It is important to ensure that the collection and accounting record process is configured to write the data to file before it is overwritten. After the results are overwritten, they are lost.

Any data not written to file is lost on a CPU switchover.

The operator can use the following **show**, **clear**, and **monitor** commands to monitor the test OAM toolset:

- The **show test-oam oam-config-summary** command provides information about the configured tests.
- The **show test-oam oam-perf** command provides the transmit (launched form me) rate information and remotely launched test receive rate on the local network element.
- The **clear test-oam oam-perf** command provides the ability to clear the test OAM performance statistics for a current view of the different rates in the preceding **oam-perf** command.
- The **monitor test-oam oam-perf** command provides time sliced performance statistics for test oam functions.

## 5 Standards and protocol support

**Note:**

The information provided in this chapter is subject to change without notice and may not apply to all platforms.

Nokia assumes no responsibility for inaccuracies.

### 5.1 Access Node Control Protocol (ANCP)

draft-ietf-ancp-protocol-02, *Protocol for Access Node Control Mechanism in Broadband Networks*

RFC 5851, *Framework and Requirements for an Access Node Control Mechanism in Broadband Multi-Service Networks*

### 5.2 Bidirectional Forwarding Detection (BFD)

draft-ietf-idr-bgp-ls-sbfd-extensions-01, *BGP Link-State Extensions for Seamless BFD*

draft-ietf-lsr-ospf-bfd-strict-mode-10, *OSPF BFD Strict-Mode*

RFC 5880, *Bidirectional Forwarding Detection (BFD)*

RFC 5881, *Bidirectional Forwarding Detection (BFD) IPv4 and IPv6 (Single Hop)*

RFC 5882, *Generic Application of Bidirectional Forwarding Detection (BFD)*

RFC 5883, *Bidirectional Forwarding Detection (BFD) for Multihop Paths*

RFC 7130, *Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces*

RFC 7880, *Seamless Bidirectional Forwarding Detection (S-BFD)*

RFC 7881, *Seamless Bidirectional Forwarding Detection (S-BFD) for IPv4, IPv6, and MPLS*

RFC 7883, *Advertising Seamless Bidirectional Forwarding Detection (S-BFD) Discriminators in IS-IS*

RFC 7884, *OSPF Extensions to Advertise Seamless Bidirectional Forwarding Detection (S-BFD) Target Discriminators*

### 5.3 Border Gateway Protocol (BGP)

draft-gredler-idr-bgplu-epe-14, *Egress Peer Engineering using BGP-LU*

draft-hares-idr-update-attr-low-bits-fix-01, *Update Attribute Flag Low Bits Clarification*

draft-ietf-idr-add-paths-guidelines-08, *Best Practices for Advertisement of Multiple Paths in IBGP*

draft-ietf-idr-best-external-03, *Advertisement of the best external route in BGP*

draft-ietf-idr-bgp-flowspec-oid-03, *Revised Validation Procedure for BGP Flow Specifications*

draft-ietf-idr-bgp-gr-notification-01, *Notification Message support for BGP Graceful Restart*  
draft-ietf-idr-bgp-ls-app-specific-attr-16, *Application-Specific Attributes Advertisement with BGP Link-State*  
draft-ietf-idr-bgp-ls-flex-algo-06, *Flexible Algorithm Definition Advertisement with BGP Link-State*  
draft-ietf-idr-bgp-optimal-route-reflection-10, *BGP Optimal Route Reflection (BGP-ORR)*  
draft-ietf-idr-error-handling-03, *Revised Error Handling for BGP UPDATE Messages*  
draft-ietf-idr-flowspec-interfaceset-03, *Applying BGP flowspec rules on a specific interface set*  
draft-ietf-idr-flowspec-path-redirect-05, *Flowspec Indirection-id Redirect – localised ID*  
draft-ietf-idr-flowspec-redirect-ip-02, *BGP Flow-Spec Redirect to IP Action*  
draft-ietf-idr-link-bandwidth-03, *BGP Link Bandwidth Extended Community*  
draft-ietf-idr-long-lived-gr-00, *Support for Long-lived BGP Graceful Restart*  
RFC 1772, *Application of the Border Gateway Protocol in the Internet*  
RFC 1997, *BGP Communities Attribute*  
RFC 2385, *Protection of BGP Sessions via the TCP MD5 Signature Option*  
RFC 2439, *BGP Route Flap Damping*  
RFC 2545, *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*  
RFC 2858, *Multiprotocol Extensions for BGP-4*  
RFC 2918, *Route Refresh Capability for BGP-4*  
RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*  
RFC 4360, *BGP Extended Communities Attribute*  
RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*  
RFC 4456, *BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)*  
RFC 4486, *Subcodes for BGP Cease Notification Message*  
RFC 4659, *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*  
RFC 4684, *Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*  
RFC 4724, *Graceful Restart Mechanism for BGP – helper mode*  
RFC 4760, *Multiprotocol Extensions for BGP-4*  
RFC 4798, *Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*  
RFC 5004, *Avoid BGP Best Path Transitions from One External to Another*  
RFC 5065, *Autonomous System Confederations for BGP*  
RFC 5291, *Outbound Route Filtering Capability for BGP-4*  
RFC 5396, *Textual Representation of Autonomous System (AS) Numbers – asplain*  
RFC 5492, *Capabilities Advertisement with BGP-4*  
RFC 5668, *4-Octet AS Specific BGP Extended Community*  
RFC 6286, *Autonomous-System-Wide Unique BGP Identifier for BGP-4*  
RFC 6793, *BGP Support for Four-Octet Autonomous System (AS) Number Space*  
RFC 6810, *The Resource Public Key Infrastructure (RPKI) to Router Protocol*

RFC 6811, *Prefix Origin Validation*  
RFC 6996, *Autonomous System (AS) Reservation for Private Use*  
RFC 7311, *The Accumulated IGP Metric Attribute for BGP*  
RFC 7606, *Revised Error Handling for BGP UPDATE Messages*  
RFC 7607, *Codification of AS 0 Processing*  
RFC 7674, *Clarification of the Flowspec Redirect Extended Community*  
RFC 7752, *North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP*  
RFC 7854, *BGP Monitoring Protocol (BMP)*  
RFC 7911, *Advertisement of Multiple Paths in BGP*  
RFC 7999, *BLACKHOLE Community*  
RFC 8092, *BGP Large Communities Attribute*  
RFC 8097, *BGP Prefix Origin Validation State Extended Community*  
RFC 8212, *Default External BGP (EBGP) Route Propagation Behavior without Policies*  
RFC 8277, *Using BGP to Bind MPLS Labels to Address Prefixes*  
RFC 8571, *BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions*  
RFC 8950, *Advertising IPv4 Network Layer Reachability Information (NLRI) with an IPv6 Next Hop*  
RFC 8955, *Dissemination of Flow Specification Rules*  
RFC 8956, *Dissemination of Flow Specification Rules for IPv6*  
RFC 9086, *Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing BGP Egress Peer Engineering*

## 5.4 Broadband Network Gateway (BNG) Control and User Plane Separation (CUPS)

3GPP TS 23.003, *Numbering, addressing and identification*  
3GPP TS 23.007, *Restoration procedures*  
3GPP TS 23.501, *System architecture for the 5G System (5GS)*  
3GPP TS 23.502, *Procedures for the 5G System (5GS)*  
3GPP TS 23.503, *Policy and charging control framework for the 5G System (5GS)*  
3GPP TS 24.501, *Non-Access-Stratum (NAS) protocol for 5G System (5GS)*  
3GPP TS 29.244, *Interface between the Control Plane and the User Plane nodes*  
3GPP TS 29.281, *General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U)*  
3GPP TS 29.500, *Technical Realization of Service Based Architecture*  
3GPP TS 29.501, *Principles and Guidelines for Services Definition*  
3GPP TS 29.502, *Session Management Services*  
3GPP TS 29.503, *Unified Data Management Services*

3GPP TS 29.512, *Session Management Policy Control Service*  
3GPP TS 29.518, *Access and Mobility Management Services*  
BBF TR-459, *Control and User Plane Separation for a Disaggregated BNG*  
BBF TR-459.2, *Multi-Service Disaggregated BNG with CUPS: Integrated Carrier Grade NAT function*  
RFC 8300, *Network Service Header (NSH)*

## 5.5 Certificate management

RFC 4210, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*  
RFC 4211, *Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)*  
RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*  
RFC 6712, *Internet X.509 Public Key Infrastructure -- HTTP Transfer for the Certificate Management Protocol (CMP)*  
RFC 7030, *Enrollment over Secure Transport*  
RFC 7468, *Textual Encodings of PKIX, PKCS, and CMS Structures*

## 5.6 Circuit emulation

RFC 4553, *Structure-Agnostic Time Division Multiplexing (TDM) over Packet (SAToP)*  
RFC 5086, *Structure-Aware Time Division Multiplexed (TDM) Circuit Emulation Service over Packet Switched Network (CESoPSN)*  
RFC 5287, *Control Protocol Extensions for the Setup of Time-Division Multiplexing (TDM) Pseudowires in MPLS Networks*

## 5.7 Ethernet

IEEE 802.1AB, *Station and Media Access Control Connectivity Discovery*  
IEEE 802.1ad, *Provider Bridges*  
IEEE 802.1ag, *Connectivity Fault Management*  
IEEE 802.1ah, *Provider Backbone Bridges*  
IEEE 802.1ak, *Multiple Registration Protocol*  
IEEE 802.1aq, *Shortest Path Bridging*  
IEEE 802.1ax, *Link Aggregation*  
IEEE 802.1D, *MAC Bridges*  
IEEE 802.1p, *Traffic Class Expediting*  
IEEE 802.1Q, *Virtual LANs*  
IEEE 802.1s, *Multiple Spanning Trees*

IEEE 802.1w, *Rapid Reconfiguration of Spanning Tree*  
IEEE 802.1X, *Port Based Network Access Control*  
IEEE 802.3ac, *VLAN Tag*  
IEEE 802.3ad, *Link Aggregation*  
IEEE 802.3ah, *Ethernet in the First Mile*  
IEEE 802.3x, *Ethernet Flow Control*  
ITU-T G.8031/Y.1342, *Ethernet Linear Protection Switching*  
ITU-T G.8032/Y.1344, *Ethernet Ring Protection Switching*  
ITU-T Y.1731, *OAM functions and mechanisms for Ethernet based networks*

## 5.8 Ethernet VPN (EVPN)

draft-ietf-bess-evpn-ipvpn-interworking-06, *EVPN Interworking with IPVPN*  
draft-ietf-bess-evpn-irb-mcast-04, *EVPN Optimized Inter-Subnet Multicast (OISM) Forwarding – ingress replication*  
draft-ietf-bess-evpn-pref-df-06, *Preference-based EVPN DF Election*  
draft-ietf-bess-evpn-unequal-lb-16, *Weighted Multi-Path Procedures for EVPN Multi-Homing – section 9*  
draft-ietf-bess-evpn-virtual-eth-segment-06, *EVPN Virtual Ethernet Segment*  
draft-ietf-bess-pbb-evpn-isid-cmacflush-00, *PBB-EVPN ISID-based CMAC-Flush*  
draft-sajassi-bess-evpn-ip-aliasing-05, *EVPN Support for L3 Fast Convergence and Aliasing/Backup Path – IP Prefix routes*  
RFC 7432, *BGP MPLS-Based Ethernet VPN*  
RFC 7623, *Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)*  
RFC 8214, *Virtual Private Wire Service Support in Ethernet VPN*  
RFC 8317, *Ethernet-Tree (E-Tree) Support in Ethernet VPN (EVPN) an Provider Backbone Bridging EVPN (PBB-EVPN)*  
RFC 8365, *A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)*  
RFC 8560, *Seamless Integration of Ethernet VPN (EVPN) with Virtual Private LAN Service (VPLS) and Their Provider Backbone Bridge (PBB) Equivalents*  
RFC 8584, *DF Election and AC-influenced DF Election*  
RFC 9047, *Propagation of ARP/ND Flags in an Ethernet Virtual Private Network (EVPN)*  
RFC 9135, *Integrated Routing and Bridging in Ethernet VPN (EVPN) – Asymmetric IRB Procedures and Mobility Procedure*  
RFC 9136, *IP Prefix Advertisement in Ethernet VPN (EVPN)*  
RFC 9161, *Operational Aspects of Proxy ARP/ND in Ethernet Virtual Private Networks*  
RFC 9251, *Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Proxies for Ethernet VPN (EVPN)*

## 5.9 gRPC Remote Procedure Calls (gRPC)

cert.proto version 0.1.0, *gRPC Network Operations Interface (gNOI) Certificate Management Service*

file.proto version 0.1.0, *gRPC Network Operations Interface (gNOI) File Service*

gnmi.proto version 0.8.0, *gRPC Network Management Interface (gNMI) Service Specification*

PROTOCOL-HTTP2, *gRPC over HTTP2*

system.proto Version 1.0.0, *gRPC Network Operations Interface (gNOI) System Service*

## 5.10 Intermediate System to Intermediate System (IS-IS)

draft-ietf-isis-mi-02, *IS-IS Multi-Instance*

draft-kaplan-isis-ext-eth-02, *Extended Ethernet Frame Size Support*

ISO/IEC 10589:2002 Second Edition, *Intermediate system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)*

RFC 1195, *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*

RFC 2973, *IS-IS Mesh Groups*

RFC 3359, *Reserved Type, Length and Value (TLV) Codepoints in Intermediate System to Intermediate System*

RFC 3719, *Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)*

RFC 3787, *Recommendations for Interoperable IP Networks using Intermediate System to Intermediate System (IS-IS)*

RFC 5120, *M-ISIS: Multi Topology (MT) Routing in IS-IS*

RFC 5130, *A Policy Control Mechanism in IS-IS Using Administrative Tags*

RFC 5301, *Dynamic Hostname Exchange Mechanism for IS-IS*

RFC 5302, *Domain-wide Prefix Distribution with Two-Level IS-IS*

RFC 5303, *Three-Way Handshake for IS-IS Point-to-Point Adjacencies*

RFC 5304, *IS-IS Cryptographic Authentication*

RFC 5305, *IS-IS Extensions for Traffic Engineering TE*

RFC 5306, *Restart Signaling for IS-IS – helper mode*

RFC 5308, *Routing IPv6 with IS-IS*

RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*

RFC 5310, *IS-IS Generic Cryptographic Authentication*

RFC 6119, *IPv6 Traffic Engineering in IS-IS*

RFC 6213, *IS-IS BFD-Enabled TLV*

RFC 6232, *Purge Originator Identification TLV for IS-IS*

RFC 6233, *IS-IS Registry Extension for Purges*  
RFC 6329, *IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging*  
RFC 7775, *IS-IS Route Preference for Extended IP and IPv6 Reachability*  
RFC 7794, *IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability* – sections 2.1 and 2.3  
RFC 7981, *IS-IS Extensions for Advertising Router Information*  
RFC 7987, *IS-IS Minimum Remaining Lifetime*  
RFC 8202, *IS-IS Multi-Instance* – single topology  
RFC 8570, *IS-IS Traffic Engineering (TE) Metric Extensions* – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE  
RFC 8919, *IS-IS Application-Specific Link Attributes*

## 5.11 Internet Protocol (IP) Fast Reroute (FRR)

draft-ietf-rtgwg-ifa-manageability-08, *Operational management of Loop Free Alternates*  
RFC 5286, *Basic Specification for IP Fast Reroute: Loop-Free Alternates*  
RFC 7431, *Multicast-Only Fast Reroute*  
RFC 7490, *Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)*  
RFC 8518, *Selection of Loop-Free Alternates for Multi-Homed Prefixes*

## 5.12 Internet Protocol (IP) general

draft-grant-tacacs-02, *The TACACS+ Protocol*  
RFC 768, *User Datagram Protocol*  
RFC 793, *Transmission Control Protocol*  
RFC 854, *Telnet Protocol Specifications*  
RFC 1350, *The TFTP Protocol (revision 2)*  
RFC 2347, *TFTP Option Extension*  
RFC 2348, *TFTP Blocksize Option*  
RFC 2349, *TFTP Timeout Interval and Transfer Size Options*  
RFC 2428, *FTP Extensions for IPv6 and NATs*  
RFC 2617, *HTTP Authentication: Basic and Digest Access Authentication*  
RFC 2784, *Generic Routing Encapsulation (GRE)*  
RFC 2818, *HTTP Over TLS*  
RFC 2890, *Key and Sequence Number Extensions to GRE*  
RFC 3164, *The BSD syslog Protocol*  
RFC 4250, *The Secure Shell (SSH) Protocol Assigned Numbers*



RFC 4251, *The Secure Shell (SSH) Protocol Architecture*  
RFC 4252, *The Secure Shell (SSH) Authentication Protocol – publickey, password*  
RFC 4253, *The Secure Shell (SSH) Transport Layer Protocol*  
RFC 4254, *The Secure Shell (SSH) Connection Protocol*  
RFC 4511, *Lightweight Directory Access Protocol (LDAP): The Protocol*  
RFC 4513, *Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms – TLS*  
RFC 4632, *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*  
RFC 5082, *The Generalized TTL Security Mechanism (GTSM)*  
RFC 5246, *The Transport Layer Security (TLS) Protocol Version 1.2 – TLS client, RSA public key*  
RFC 5425, *Transport Layer Security (TLS) Transport Mapping for Syslog – RFC 3164 with TLS*  
RFC 5656, *Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer – ECDSA*  
RFC 5925, *The TCP Authentication Option*  
RFC 5926, *Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)*  
RFC 6398, *IP Router Alert Considerations and Usage – MLD*  
RFC 6528, *Defending against Sequence Number Attacks*  
RFC 7011, *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information*  
RFC 7012, *Information Model for IP Flow Information Export*  
RFC 7230, *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*  
RFC 7231, *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*  
RFC 7232, *Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests*  
RFC 7301, *Transport Layer Security (TLS) Application Layer Protocol Negotiation Extension*  
RFC 7616, *HTTP Digest Access Authentication*  
RFC 8446, *The Transport Layer Security (TLS) Protocol Version 1.3*

## 5.13 Internet Protocol (IP) multicast

cisco-ipmulticast/pim-autorp-spec01, *Auto-RP: Automatic discovery of Group-to-RP mappings for IP multicast – version 1*  
draft-ietf-bier-pim-signaling-08, *PIM Signaling Through BIER Core*  
draft-ietf-idmr-traceroute-ipm-07, *A "traceroute" facility for IP Multicast*  
draft-ietf-l2vpn-vpls-pim-snooping-07, *Protocol Independent Multicast (PIM) over Virtual Private LAN Service (VPLS)*  
RFC 1112, *Host Extensions for IP Multicasting*  
RFC 2236, *Internet Group Management Protocol, Version 2*  
RFC 2365, *Administratively Scoped IP Multicast*

RFC 2375, *IPv6 Multicast Address Assignments*

RFC 2710, *Multicast Listener Discovery (MLD) for IPv6*

RFC 3306, *Unicast-Prefix-based IPv6 Multicast Addresses*

RFC 3376, *Internet Group Management Protocol, Version 3*

RFC 3446, *Anycast Rendezvous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)*

RFC 3590, *Source Address Selection for the Multicast Listener Discovery (MLD) Protocol*

RFC 3618, *Multicast Source Discovery Protocol (MSDP)*

RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*

RFC 3956, *Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*

RFC 3973, *Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised) – auto-RP groups*

RFC 4541, *Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches*

RFC 4604, *Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast*

RFC 4607, *Source-Specific Multicast for IP*

RFC 4608, *Source-Specific Protocol Independent Multicast in 232/8*

RFC 4610, *Anycast-RP Using Protocol Independent Multicast (PIM)*

RFC 4611, *Multicast Source Discovery Protocol (MSDP) Deployment Scenarios*

RFC 5059, *Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*

RFC 5186, *Internet Group Management Protocol Version 3 (IGMPv3) / Multicast Listener Discovery Version 2 (MLDv2) and Multicast Routing Protocol Interaction*

RFC 5384, *The Protocol Independent Multicast (PIM) Join Attribute Format*

RFC 5496, *The Reverse Path Forwarding (RPF) Vector TLV*

RFC 6037, *Cisco Systems' Solution for Multicast in MPLS/BGP IP VPNs*

RFC 6512, *Using Multipoint LDP When the Backbone Has No Route to the Root*

RFC 6513, *Multicast in MPLS/BGP IP VPNs*

RFC 6514, *BGP Encodings and Procedures for Multicast in MPLS/IP VPNs*

RFC 6515, *IPv4 and IPv6 Infrastructure Addresses in BGP Updates for Multicast VPNs*

RFC 6516, *IPv6 Multicast VPN (MVPN) Support Using PIM Control Plane and Selective Provider Multicast Service Interface (S-PMSI) Join Messages*

RFC 6625, *Wildcards in Multicast VPN Auto-Discover Routes*

RFC 6826, *Multipoint LDP In-Band Signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Path*

RFC 7246, *Multipoint Label Distribution Protocol In-Band Signaling in a Virtual Routing and Forwarding (VRF) Table Context*

RFC 7385, *IANA Registry for P-Multicast Service Interface (PMSI) Tunnel Type Code Points*

RFC 7716, *Global Table Multicast with BGP Multicast VPN (BGP-MVPN) Procedures*

RFC 7761, *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*  
RFC 8279, *Multicast Using Bit Index Explicit Replication (BIER)*  
RFC 8296, *Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks – MPLS encapsulation*  
RFC 8401, *Bit Index Explicit Replication (BIER) Support via IS-IS*  
RFC 8444, *OSPFv2 Extensions for Bit Index Explicit Replication (BIER)*  
RFC 8487, *Mtrace Version 2: Traceroute Facility for IP Multicast*  
RFC 8534, *Explicit Tracking with Wildcard Routes in Multicast VPN – (C-\*,C-\*) wildcard*  
RFC 8556, *Multicast VPN Using Bit Index Explicit Replication (BIER)*

## 5.14 Internet Protocol (IP) version 4

RFC 791, *Internet Protocol*  
RFC 792, *Internet Control Message Protocol*  
RFC 826, *An Ethernet Address Resolution Protocol*  
RFC 951, *Bootstrap Protocol (BOOTP) – relay*  
RFC 1034, *Domain Names - Concepts and Facilities*  
RFC 1035, *Domain Names - Implementation and Specification*  
RFC 1191, *Path MTU Discovery – router specification*  
RFC 1519, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*  
RFC 1534, *Interoperation between DHCP and BOOTP*  
RFC 1542, *Clarifications and Extensions for the Bootstrap Protocol*  
RFC 1812, *Requirements for IPv4 Routers*  
RFC 1918, *Address Allocation for Private Internets*  
RFC 2003, *IP Encapsulation within IP*  
RFC 2131, *Dynamic Host Configuration Protocol*  
RFC 2132, *DHCP Options and BOOTP Vendor Extensions*  
RFC 2401, *Security Architecture for Internet Protocol*  
RFC 3021, *Using 31-Bit Prefixes on IPv4 Point-to-Point Links*  
RFC 3046, *DHCP Relay Agent Information Option (Option 82)*  
RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*  
RFC 4884, *Extended ICMP to Support Multi-Part Messages – ICMPv4 and ICMPv6 Time Exceeded*

## 5.15 Internet Protocol (IP) version 6

RFC 2464, *Transmission of IPv6 Packets over Ethernet Networks*

---

RFC 2529, *Transmission of IPv6 over IPv4 Domains without Explicit Tunnels*  
RFC 3122, *Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification*  
RFC 3315, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*  
RFC 3587, *IPv6 Global Unicast Address Format*  
RFC 3596, *DNS Extensions to Support IP version 6*  
RFC 3633, *IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6*  
RFC 3646, *DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*  
RFC 3736, *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6*  
RFC 3971, *SEcure Neighbor Discovery (SEND)*  
RFC 3972, *Cryptographically Generated Addresses (CGA)*  
RFC 4007, *IPv6 Scoped Address Architecture*  
RFC 4193, *Unique Local IPv6 Unicast Addresses*  
RFC 4291, *Internet Protocol Version 6 (IPv6) Addressing Architecture*  
RFC 4443, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*  
RFC 4861, *Neighbor Discovery for IP version 6 (IPv6)*  
RFC 4862, *IPv6 Stateless Address Autoconfiguration – router functions*  
RFC 4890, *Recommendations for Filtering ICMPv6 Messages in Firewalls*  
RFC 4941, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*  
RFC 5007, *DHCPv6 Leasequery*  
RFC 5095, *Deprecation of Type 0 Routing Headers in IPv6*  
RFC 5722, *Handling of Overlapping IPv6 Fragments*  
RFC 5798, *Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6 – IPv6*  
RFC 5952, *A Recommendation for IPv6 Address Text Representation*  
RFC 6092, *Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service – Internet Control and Management, Upper-Layer Transport Protocols, UDP Filters, IPsec and Internet Key Exchange (IKE), TCP Filters*  
RFC 6106, *IPv6 Router Advertisement Options for DNS Configuration*  
RFC 6164, *Using 127-Bit IPv6 Prefixes on Inter-Router Links*  
RFC 6437, *IPv6 Flow Label Specification*  
RFC 6603, *Prefix Exclude Option for DHCPv6-based Prefix Delegation*  
RFC 8021, *Generation of IPv6 Atomic Fragments Considered Harmful*  
RFC 8200, *Internet Protocol, Version 6 (IPv6) Specification*  
RFC 8201, *Path MTU Discovery for IP version 6*

## 5.16 Internet Protocol Security (IPsec)

draft-ietf-ipsec-isakmp-mode-cfg-05, *The ISAKMP Configuration Method*  
draft-ietf-ipsec-isakmp-xauth-06, *Extended Authentication within ISAKMP/Oakley (XAUTH)*  
RFC 2401, *Security Architecture for the Internet Protocol*  
RFC 2403, *The Use of HMAC-MD5-96 within ESP and AH*  
RFC 2404, *The Use of HMAC-SHA-1-96 within ESP and AH*  
RFC 2405, *The ESP DES-CBC Cipher Algorithm With Explicit IV*  
RFC 2406, *IP Encapsulating Security Payload (ESP)*  
RFC 2407, *IPsec Domain of Interpretation for ISAKMP (IPsec DoI)*  
RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)*  
RFC 2409, *The Internet Key Exchange (IKE)*  
RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec*  
RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*  
RFC 3526, *More Modular Exponential (MODP) Diffie-Hellman group for Internet Key Exchange (IKE)*  
RFC 3566, *The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec*  
RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*  
RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*  
RFC 3947, *Negotiation of NAT-Traversal in the IKE*  
RFC 3948, *UDP Encapsulation of IPsec ESP Packets*  
RFC 4106, *The Use of Galois/Counter Mode (GCM) in IPsec ESP*  
RFC 4109, *Algorithms for Internet Key Exchange version 1 (IKEv1)*  
RFC 4301, *Security Architecture for the Internet Protocol*  
RFC 4303, *IP Encapsulating Security Payload*  
RFC 4307, *Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)*  
RFC 4308, *Cryptographic Suites for IPsec*  
RFC 4434, *The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)*  
RFC 4543, *The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH*  
RFC 4754, *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*  
RFC 4835, *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*  
RFC 4868, *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*  
RFC 4945, *The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2 and PKIX*  
RFC 5019, *The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments*  
RFC 5282, *Using Authenticated Encryption Algorithms with the Encrypted Payload of the IKEv2 Protocol*  
RFC 5903, *ECP Groups for IKE and IKEv2*

RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)*  
RFC 5998, *An Extension for EAP-Only Authentication in IKEv2*  
RFC 6379, *Suite B Cryptographic Suites for IPsec*  
RFC 6380, *Suite B Profile for Internet Protocol Security (IPsec)*  
RFC 6960, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*  
RFC 7296, *Internet Key Exchange Protocol Version 2 (IKEv2)*  
RFC 7321, *Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)*  
RFC 7383, *Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation*  
RFC 7427, *Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)*

## 5.17 Label Distribution Protocol (LDP)

draft-pdutta-mpls-ldp-adj-capability-00, *LDP Adjacency Capabilities*  
draft-pdutta-mpls-ldp-v2-00, *LDP Version 2*  
draft-pdutta-mpls-mldp-up-redundancy-00, *Upstream LSR Redundancy for Multi-point LDP Tunnels*  
draft-pdutta-mpls-multi-ldp-instance-00, *Multiple LDP Instances*  
draft-pdutta-mpls-tldp-hello-reduce-04, *Targeted LDP Hello Reduction*  
RFC 3037, *LDP Applicability*  
RFC 3478, *Graceful Restart Mechanism for Label Distribution Protocol – helper mode*  
RFC 5036, *LDP Specification*  
RFC 5283, *LDP Extension for Inter-Area Label Switched Paths (LSPs)*  
RFC 5443, *LDP IGP Synchronization*  
RFC 5561, *LDP Capabilities*  
RFC 5919, *Signaling LDP Label Advertisement Completion*  
RFC 6388, *Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*  
RFC 6512, *Using Multipoint LDP When the Backbone Has No Route to the Root*  
RFC 6826, *Multipoint LDP in-band signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*  
RFC 7032, *LDP Downstream-on-Demand in Seamless MPLS*  
RFC 7473, *Controlling State Advertisements of Non-negotiated LDP Applications*  
RFC 7552, *Updates to LDP for IPv6*

## 5.18 Layer Two Tunneling Protocol (L2TP) Network Server (LNS)

draft-mammoliti-l2tp-accessline-avp-04, *Layer 2 Tunneling Protocol (L2TP) Access Line Information Attribute Value Pair (AVP) Extensions*

RFC 2661, *Layer Two Tunneling Protocol "L2TP"*

RFC 2809, *Implementation of L2TP Compulsory Tunneling via RADIUS*

RFC 3438, *Layer Two Tunneling Protocol (L2TP) Internet Assigned Numbers: Internet Assigned Numbers Authority (IANA) Considerations Update*

RFC 3931, *Layer Two Tunneling Protocol - Version 3 (L2TPv3)*

RFC 4719, *Transport of Ethernet Frames over Layer 2 Tunneling Protocol Version 3 (L2TPv3)*

RFC 4951, *Fail Over Extensions for Layer 2 Tunneling Protocol (L2TP) "failover"*

## 5.19 Multiprotocol Label Switching (MPLS)

draft-ietf-mpls-lsp-ping-ospfv3-codepoint-02, *OSPFv3 CodePoint for MPLS LSP Ping*

RFC 3031, *Multiprotocol Label Switching Architecture*

RFC 3032, *MPLS Label Stack Encoding*

RFC 3270, *Multi-Protocol Label Switching (MPLS) Support of Differentiated Services – E-LSP*

RFC 3443, *Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks*

RFC 4023, *Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)*

RFC 4182, *Removing a Restriction on the use of MPLS Explicit NULL*

RFC 4950, *ICMP Extensions for Multiprotocol Label Switching*

RFC 5332, *MPLS Multicast Encapsulations*

RFC 5884, *Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)*

RFC 6374, *Packet Loss and Delay Measurement for MPLS Networks – Delay Measurement, Channel Type 0x000C*

RFC 6424, *Mechanism for Performing Label Switched Path Ping (LSP Ping) over MPLS Tunnels*

RFC 6425, *Detecting Data Plane Failures in Point-to-Multipoint Multiprotocol Label Switching (MPLS) - Extensions to LSP Ping*

RFC 6790, *The Use of Entropy Labels in MPLS Forwarding*

RFC 7308, *Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)*

RFC 7510, *Encapsulating MPLS in UDP*

RFC 7746, *Label Switched Path (LSP) Self-Ping*

RFC 7876, *UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks – Delay Measurement*

RFC 8029, *Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures*

## 5.20 Multiprotocol Label Switching - Transport Profile (MPLS-TP)

RFC 5586, *MPLS Generic Associated Channel*

RFC 5921, *A Framework for MPLS in Transport Networks*

RFC 5960, *MPLS Transport Profile Data Plane Architecture*

RFC 6370, *MPLS Transport Profile (MPLS-TP) Identifiers*

RFC 6378, *MPLS Transport Profile (MPLS-TP) Linear Protection*

RFC 6426, *MPLS On-Demand Connectivity and Route Tracing*

RFC 6427, *MPLS Fault Management Operations, Administration, and Maintenance (OAM)*

RFC 6428, *Proactive Connectivity Verification, Continuity Check and Remote Defect indication for MPLS Transport Profile*

RFC 6478, *Pseudowire Status for Static Pseudowires*

RFC 7213, *MPLS Transport Profile (MPLS-TP) Next-Hop Ethernet Addressing*

## 5.21 Network Address Translation (NAT)

draft-ietf-behave-address-format-10, *IPv6 Addressing of IPv4/IPv6 Translators*

draft-ietf-behave-v6v4-xlate-23, *IP/ICMP Translation Algorithm*

draft-miles-behave-l2nat-00, *Layer2-Aware NAT*

draft-nishitani-cgn-02, *Common Functions of Large Scale NAT (LSN)*

RFC 4787, *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*

RFC 5382, *NAT Behavioral Requirements for TCP*

RFC 5508, *NAT Behavioral Requirements for ICMP*

RFC 6146, *Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers*

RFC 6333, *Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion*

RFC 6334, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite*

RFC 6887, *Port Control Protocol (PCP)*

RFC 6888, *Common Requirements For Carrier-Grade NATs (CGNs)*

RFC 7753, *Port Control Protocol (PCP) Extension for Port-Set Allocation*

RFC 7915, *IP/ICMP Translation Algorithm*

## 5.22 Network Configuration Protocol (NETCONF)

RFC 5277, *NETCONF Event Notifications*

RFC 6020, *YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)*

RFC 6022, *YANG Module for NETCONF Monitoring*

RFC 6241, *Network Configuration Protocol (NETCONF)*



RFC 6242, *Using the NETCONF Protocol over Secure Shell (SSH)*  
RFC 6243, *With-defaults Capability for NETCONF*  
RFC 8342, *Network Management Datastore Architecture (NMDA) – Startup, Candidate, Running and Intended datastores*  
RFC 8525, *YANG Library*  
RFC 8526, *NETCONF Extensions to Support the Network Management Datastore Architecture – <get-data> operation*

## 5.23 Open Shortest Path First (OSPF)

RFC 1765, *OSPF Database Overflow*  
RFC 2328, *OSPF Version 2*  
RFC 3101, *The OSPF Not-So-Stubby Area (NSSA) Option*  
RFC 3509, *Alternative Implementations of OSPF Area Border Routers*  
RFC 3623, *Graceful OSPF Restart Graceful OSPF Restart – helper mode*  
RFC 3630, *Traffic Engineering (TE) Extensions to OSPF Version 2*  
RFC 4222, *Prioritized Treatment of Specific OSPF Version 2 Packets and Congestion Avoidance*  
RFC 4552, *Authentication/Confidentiality for OSPFv3*  
RFC 4576, *Using a Link State Advertisement (LSA) Options Bit to Prevent Looping in BGP/MPLS IP Virtual Private Networks (VPNs)*  
RFC 4577, *OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*  
RFC 5185, *OSPF Multi-Area Adjacency*  
RFC 5187, *OSPFv3 Graceful Restart – helper mode*  
RFC 5243, *OSPF Database Exchange Summary List Optimization*  
RFC 5250, *The OSPF Opaque LSA Option*  
RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*  
RFC 5340, *OSPF for IPv6*  
RFC 5642, *Dynamic Hostname Exchange Mechanism for OSPF*  
RFC 5709, *OSPFv2 HMAC-SHA Cryptographic Authentication*  
RFC 5838, *Support of Address Families in OSPFv3*  
RFC 6549, *OSPFv2 Multi-Instance Extensions*  
RFC 6987, *OSPF Stub Router Advertisement*  
RFC 7471, *OSPF Traffic Engineering (TE) Metric Extensions – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE*  
RFC 7684, *OSPFv2 Prefix/Link Attribute Advertisement*  
RFC 7770, *Extensions to OSPF for Advertising Optional Router Capabilities*  
RFC 8362, *OSPFv3 Link State Advertisement (LSA) Extensibility*

RFC 8920, *OSPF Application-Specific Link Attributes*

## 5.24 OpenFlow

TS-007 Version 1.3.1, *OpenFlow Switch Specification* – OpenFlow-hybrid switches

## 5.25 Path Computation Element Protocol (PCEP)

draft-ietf-pce-binding-label-sid-15, *Carrying Binding Label/Segment Identifier (SID) in PCE-based Networks*. – MPLS binding SIDs

draft-alvarez-pce-path-profiles-04, *PCE Path Profiles*

draft-dhs-spring-pce-sr-p2mp-policy-00, *PCEP extensions for p2mp sr policy*

RFC 5440, *Path Computation Element (PCE) Communication Protocol (PCEP)*

RFC 8231, *Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE*

RFC 8253, *PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)*

RFC 8281, *PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model*

RFC 8408, *Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages*

RFC 8664, *Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing*

## 5.26 Point-to-Point Protocol (PPP)

RFC 1332, *The PPP Internet Protocol Control Protocol (IPCP)*

RFC 1990, *The PPP Multilink Protocol (MP)*

RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*

RFC 2516, *A Method for Transmitting PPP Over Ethernet (PPPoE)*

RFC 4638, *Accommodating a Maximum Transit Unit/Maximum Receive Unit (MTU/MRU) Greater Than 1492 in the Point-to-Point Protocol over Ethernet (PPPoE)*

RFC 5072, *IP Version 6 over PPP*

## 5.27 Policy management and credit control

3GPP TS 29.212 Release 11, *Policy and Charging Control (PCC); Reference points* – Gx support as it applies to wireline environment (BNG)

RFC 4006, *Diameter Credit-Control Application*

RFC 6733, *Diameter Base Protocol*

## 5.28 Pseudowire (PW)

draft-ietf-l2vpn-vpws-iw-oam-04, *OAM Procedures for VPWS Interworking*  
MFA Forum 12.0.0, *Multiservice Interworking - Ethernet over MPLS*  
MFA Forum 13.0.0, *Fault Management for Multiservice Interworking v1.0*  
MFA Forum 16.0.0, *Multiservice Interworking - IP over MPLS*  
RFC 3916, *Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)*  
RFC 3985, *Pseudo Wire Emulation Edge-to-Edge (PWE3)*  
RFC 4385, *Pseudo Wire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN*  
RFC 4446, *IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)*  
RFC 4447, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*  
RFC 4448, *Encapsulation Methods for Transport of Ethernet over MPLS Networks*  
RFC 5085, *Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires*  
RFC 5659, *An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge*  
RFC 5885, *Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)*  
RFC 6073, *Segmented Pseudowire*  
RFC 6310, *Pseudowire (PW) Operations, Administration, and Maintenance (OAM) Message Mapping*  
RFC 6391, *Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network*  
RFC 6575, *Address Resolution Protocol (ARP) Mediation for IP Interworking of Layer 2 VPNs*  
RFC 6718, *Pseudowire Redundancy*  
RFC 6829, *Label Switched Path (LSP) Ping for Pseudowire Forwarding Equivalence Classes (FECs) Advertised over IPv6*  
RFC 6870, *Pseudowire Preferential Forwarding Status bit*  
RFC 7023, *MPLS and Ethernet Operations, Administration, and Maintenance (OAM) Interworking*  
RFC 7267, *Dynamic Placement of Multi-Segment Pseudowires*  
RFC 7392, *Explicit Path Routing for Dynamic Multi-Segment Pseudowires – ER-TLV and ER-HOP IPv4 Prefix*  
RFC 8395, *Extensions to BGP-Signaled Pseudowires to Support Flow-Aware Transport Labels*

## 5.29 Quality of Service (QoS)

RFC 2430, *A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)*  
RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*  
RFC 2597, *Assured Forwarding PHB Group*  
RFC 3140, *Per Hop Behavior Identification Codes*  
RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*

## 5.30 Remote Authentication Dial In User Service (RADIUS)

draft-oscca-cfrg-sm3-02, *The SM3 Cryptographic Hash Function*  
RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*  
RFC 2866, *RADIUS Accounting*  
RFC 2867, *RADIUS Accounting Modifications for Tunnel Protocol Support*  
RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*  
RFC 2869, *RADIUS Extensions*  
RFC 3162, *RADIUS and IPv6*  
RFC 4818, *RADIUS Delegated-IPv6-Prefix Attribute*  
RFC 5176, *Dynamic Authorization Extensions to RADIUS*  
RFC 6613, *RADIUS over TCP – with TLS*  
RFC 6614, *Transport Layer Security (TLS) Encryption for RADIUS*  
RFC 6929, *Remote Authentication Dial-In User Service (RADIUS) Protocol Extensions*  
RFC 6911, *RADIUS attributes for IPv6 Access Networks*

## 5.31 Resource Reservation Protocol - Traffic Engineering (RSVP-TE)

draft-newton-mpls-te-dynamic-overbooking-00, *A Diffserv-TE Implementation Model to dynamically change booking factors during failure events*  
RFC 2702, *Requirements for Traffic Engineering over MPLS*  
RFC 2747, *RSVP Cryptographic Authentication*  
RFC 2961, *RSVP Refresh Overhead Reduction Extensions*  
RFC 3097, *RSVP Cryptographic Authentication -- Updated Message Type Value*  
RFC 3209, *RSVP-TE: Extensions to RSVP for LSP Tunnels*  
RFC 3477, *Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)*  
RFC 3564, *Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering*  
RFC 3906, *Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels*  
RFC 4090, *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*  
RFC 4124, *Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering*  
RFC 4125, *Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*  
RFC 4127, *Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*  
RFC 4561, *Definition of a Record Route Object (RRO) Node-Id Sub-Object*  
RFC 4875, *Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)*  
RFC 5712, *MPLS Traffic Engineering Soft Preemption*

RFC 5817, *Graceful Shutdown in MPLS and Generalized MPLS Traffic Engineering Networks*

## 5.32 Routing Information Protocol (RIP)

RFC 1058, *Routing Information Protocol*

RFC 2080, *RIPng for IPv6*

RFC 2082, *RIP-2 MD5 Authentication*

RFC 2453, *RIP Version 2*

## 5.33 Segment Routing (SR)

draft-bashandy-rtgwg-segment-routing-uloop-06, *Loop avoidance using Segment Routing*

draft-filsfils-spring-net-pgm-extension-srv6-usid-13, *Network Programming extension: SRv6 uSID instruction*

draft-filsfils-spring-srv6-net-pgm-insertion-04, *SRv6 NET-PGM extension: Insertion*

draft-ietf-6man-spring-srv6-oam-10, *Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)*

draft-ietf-idr-bgp-ls-segment-routing-ext-16, *BGP Link-State extensions for Segment Routing*

draft-ietf-idr-bgp-ls-srv6-ext-13, *BGP Link State Extensions for SRv6*

draft-ietf-idr-segment-routing-te-policy-11, *Advertising Segment Routing Policies in BGP*

draft-ietf-isis-mpls-enc-10, *Signaling Entropy Label Capability and Entropy Readable Label Depth Using IS-IS – advertising ELC*

draft-ietf-lsr-flex-algo-16, *IGP Flexible Algorithm*

draft-ietf-lsr-isis-srv6-extensions-14, *IS-IS Extension to Support Segment Routing over IPv6 Dataplane*

draft-ietf-ospf-mpls-enc-12, *Signaling Entropy Label Capability and Entropy Readable Label-stack Depth Using OSPF – advertising ELC*

draft-ietf-rtgwg-segment-routing-ti-lfa-01, *Topology Independent Fast Reroute using Segment Routing*

draft-ietf-spring-conflict-resolution-05, *Segment Routing MPLS Conflict Resolution*

draft-ietf-teas-sr-rsvp-coexistence-rec-02, *Recommendations for RSVP-TE and Segment Routing LSP co-existence*

draft-voyer-6man-extension-header-insertion-10, *Deployments With Insertion of IPv6 Segment Routing Headers*

draft-voyer-pim-sr-p2mp-policy-02, *Segment Routing Point-to-Multipoint Policy*

draft-voyer-spring-sr-p2mp-policy-03, *SR Replication Policy for P2MP Service Delivery*

RFC 8287, *Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes*

RFC 8476, *Signaling Maximum SID Depth (MSD) Using OSPF – node MSD*

RFC 8491, *Signaling Maximum SID Depth (MSD) Using IS-IS – node MSD*

RFC 8660, *Segment Routing with the MPLS Data Plane*  
RFC 8661, *Segment Routing MPLS Interworking with LDP*  
RFC 8663, *MPLS Segment Routing over IP – BGP SR with SR-MPLS-over-UDP/IP*  
RFC 8665, *OSPF Extensions for Segment Routing*  
RFC 8666, *OSPFv3 Extensions for Segment Routing*  
RFC 8667, *IS-IS Extensions for Segment Routing*  
RFC 8669, *Segment Routing Prefix Segment Identifier Extensions for BGP*  
RFC 8754, *IPv6 Segment Routing Header (SRH)*  
RFC 8814, *Signaling Maximum SID Depth (MSD) Using the Border Gateway Protocol - Link State*  
RFC 8986, *Segment Routing over IPv6 (SRv6) Network Programming*  
RFC 9252, *BGP Overlay Services Based on Segment Routing over IPv6 (SRv6)*  
RFC 9256, *Segment Routing Policy Architecture*

## 5.34 Simple Network Management Protocol (SNMP)

draft-blumenthal-aes-usm-04, *The AES Cipher Algorithm in the SNMP's User-based Security Model – CFB128-AES-192 and CFB128-AES-256*  
draft-ietf-isis-wg-mib-06, *Management Information Base for Intermediate System to Intermediate System (IS-IS)*  
draft-ietf-mboned-msdp-mib-01, *Multicast Source Discovery protocol MIB*  
draft-ietf-mpls-ldp-mib-07, *Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)*  
draft-ietf-mpls-lsr-mib-06, *Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base Using SMIv2*  
draft-ietf-mpls-te-mib-04, *Multiprotocol Label Switching (MPLS) Traffic Engineering Management Information Base*  
draft-ietf-ospf-mib-update-08, *OSPF Version 2 Management Information Base*  
draft-ietf-rrp-unified-mib-06, *Definitions of Managed Objects for the VRRP over IPv4 and IPv6 – IPv6*  
ESO-CONSORTIUM-MIB revision 200406230000Z, *esoConsortiumMIB*  
IANA-ADDRESS-FAMILY-NUMBERS-MIB revision 200203140000Z, *ianaAddressFamilyNumbers*  
IANAifType-MIB revision 200505270000Z, *ianaifType*  
IANA-RTPROTO-MIB revision 200009260000Z, *ianaRtProtoMIB*  
IEEE8021-CFM-MIB revision 200706100000Z, *ieee8021CfmMib*  
IEEE8021-PAE-MIB revision 200101160000Z, *ieee8021paeMIB*  
IEEE8023-LAG-MIB revision 200006270000Z, *lagMIB*  
LLDP-MIB revision 200505060000Z, *lldpMIB*  
RFC 1157, *A Simple Network Management Protocol (SNMP)*  
RFC 1212, *Concise MIB Definitions*

---

RFC 1215, *A Convention for Defining Traps for use with the SNMP*

RFC 1724, *RIP Version 2 MIB Extension*

RFC 1901, *Introduction to Community-based SNMPv2*

RFC 2021, *Remote Network Monitoring Management Information Base Version 2 using SMIv2*

RFC 2206, *RSVP Management Information Base using SMIv2*

RFC 2213, *Integrated Services Management Information Base using SMIv2*

RFC 2494, *Definitions of Managed Objects for the DS0 and DS0 Bundle Interface Type*

RFC 2578, *Structure of Management Information Version 2 (SMIv2)*

RFC 2579, *Textual Conventions for SMIv2*

RFC 2580, *Conformance Statements for SMIv2*

RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*

RFC 2819, *Remote Network Monitoring Management Information Base*

RFC 2856, *Textual Conventions for Additional High Capacity Data Types*

RFC 2863, *The Interfaces Group MIB*

RFC 2864, *The Inverted Stack Table Extension to the Interfaces Group MIB*

RFC 2933, *Internet Group Management Protocol MIB*

RFC 3014, *Notification Log MIB*

RFC 3165, *Definitions of Managed Objects for the Delegation of Management Scripts*

RFC 3231, *Definitions of Managed Objects for Scheduling Management Operations*

RFC 3273, *Remote Network Monitoring Management Information Base for High Capacity Networks*

RFC 3410, *Introduction and Applicability Statements for Internet Standard Management Framework*

RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*

RFC 3412, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*

RFC 3413, *Simple Network Management Protocol (SNMP) Applications*

RFC 3414, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*

RFC 3415, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*

RFC 3416, *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*

RFC 3417, *Transport Mappings for the Simple Network Management Protocol (SNMP) – SNMP over UDP over IPv4*

RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*

RFC 3419, *Textual Conventions for Transport Addresses*

RFC 3498, *Definitions of Managed Objects for Synchronous Optical Network (SONET) Linear Automatic Protection Switching (APS) Architectures*

RFC 3584, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*

RFC 3592, *Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type*

RFC 3593, *Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals*

RFC 3635, *Definitions of Managed Objects for the Ethernet-like Interface Types*

RFC 3637, *Definitions of Managed Objects for the Ethernet WAN Interface Sublayer*

RFC 3826, *The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model*

RFC 3877, *Alarm Management Information Base (MIB)*

RFC 3895, *Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Types*

RFC 3896, *Definitions of Managed Objects for the DS3/E3 Interface Type*

RFC 4001, *Textual Conventions for Internet Network Addresses*

RFC 4022, *Management Information Base for the Transmission Control Protocol (TCP)*

RFC 4113, *Management Information Base for the User Datagram Protocol (UDP)*

RFC 4220, *Traffic Engineering Link Management Information Base*

RFC 4273, *Definitions of Managed Objects for BGP-4*

RFC 4292, *IP Forwarding Table MIB*

RFC 4293, *Management Information Base for the Internet Protocol (IP)*

RFC 4631, *Link Management Protocol (LMP) Management Information Base (MIB)*

RFC 4878, *Definitions and Managed Objects for Operations, Administration, and Maintenance (OAM) Functions on Ethernet-Like Interfaces*

RFC 7420, *Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module*

RFC 7630, *HMAC-SHA-2 Authentication Protocols in the User-based Security Model (USM) for SNMPv3*

SFLOW-MIB revision 200309240000Z, *sFlowMIB*

## 5.35 Timing

GR-1244-CORE Issue 3, *Clocks for the Synchronized Network: Common Generic Criteria*

GR-253-CORE Issue 3, *SONET Transport Systems: Common Generic Criteria*

IEEE 1588-2008, *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*

ITU-T G.781, *Synchronization layer functions*

ITU-T G.813, *Timing characteristics of SDH equipment slave clocks (SEC)*

ITU-T G.8261, *Timing and synchronization aspects in packet networks*

ITU-T G.8262, *Timing characteristics of synchronous Ethernet equipment slave clock (EEC)*

ITU-T G.8262.1, *Timing characteristics of an enhanced synchronous Ethernet equipment slave clock (eEEC)*

ITU-T G.8264, *Distribution of timing information through packet networks*



ITU-T G.8265.1, *Precision time protocol telecom profile for frequency synchronization*

ITU-T G.8275.1, *Precision time protocol telecom profile for phase/time synchronization with full timing support from the network*

RFC 3339, *Date and Time on the Internet: Timestamps*

RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*

### 5.36 Two-Way Active Measurement Protocol (TWAMP)

RFC 5357, *A Two-Way Active Measurement Protocol (TWAMP) – server, unauthenticated mode*

RFC 5938, *Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)*

RFC 6038, *Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features*

RFC 8545, *Well-Known Port Assignments for the One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP) – TWAMP*

RFC 8762, *Simple Two-Way Active Measurement Protocol – unauthenticated*

RFC 8972, *Simple Two-Way Active Measurement Protocol Optional Extensions – unauthenticated*

### 5.37 Virtual Private LAN Service (VPLS)

RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*

RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*

RFC 5501, *Requirements for Multicast Support in Virtual Private LAN Services*

RFC 6074, *Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)*

RFC 7041, *Extensions to the Virtual Private LAN Service (VPLS) Provider Edge (PE) Model for Provider Backbone Bridging*

RFC 7117, *Multicast in Virtual Private LAN Service (VPLS)*

### 5.38 Voice and video

DVB BlueBook A86, *Transport of MPEG-2 TS Based DVB Services over IP Based Networks*

ETSI TS 101 329-5 Annex E, *QoS Measurement for VoIP - Method for determining an Equipment Impairment Factor using Passive Monitoring*

ITU-T G.1020 Appendix I, *Performance Parameter Definitions for Quality of Speech and other Voiceband Applications Utilizing IP Networks - Mean Absolute Packet Delay Variation & Markov Models*

ITU-T G.107, *The E Model - A computational model for use in planning*

ITU-T P.564, *Conformance testing for voice over IP transmission quality assessment models*

RFC 3550, *RTP: A Transport Protocol for Real-Time Applications – Appendix A.8*

RFC 4585, *Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)*

RFC 4588, *RTP Retransmission Payload Format*

## 5.39 Wireless Local Area Network (WLAN) gateway

3GPP TS 23.402, *Architecture enhancements for non-3GPP accesses – S2a roaming based on GPRS*

## 5.40 Yet Another Next Generation (YANG)

RFC 6991, *Common YANG Data Types*

RFC 7950, *The YANG 1.1 Data Modeling Language*

RFC 7951, *JSON Encoding of Data Modeled with YANG*

## 5.41 Yet Another Next Generation (YANG) OpenConfig Modules

openconfig-aaa.yang version 0.4.0, *OpenConfig AAA Module*

openconfig-aaa-radius.yang version 0.3.0, *OpenConfig AAA RADIUS Module*

openconfig-aaa-tacacs.yang version 0.3.0, *OpenConfig AAA TACACS+ Module*

openconfig-acl.yang version 1.0.0, *OpenConfig ACL Module*

openconfig-bfd.yang version 0.2.2, *OpenConfig BFD Module*

openconfig-bgp.yang version 6.1.0, *OpenConfig BGP Module*

openconfig-bgp-common.yang version 6.0.0, *OpenConfig BGP Common Module*

openconfig-bgp-common-multiprotocol.yang version 6.0.0, *OpenConfig BGP Common Multiprotocol Module*

openconfig-bgp-common-structure.yang version 6.0.0, *OpenConfig BGP Common Structure Module*

openconfig-bgp-global.yang version 6.0.0 *OpenConfig BGP Global Module*

openconfig-bgp-neighbor.yang version 6.1.0, *OpenConfig BGP Neighbor Module*

openconfig-bgp-peer-group.yang version 6.1.0, *OpenConfig BGP Peer Group Module*

openconfig-bgp-policy.yang version 4.0.1, *OpenConfig BGP Policy Module*

openconfig-if-aggregate.yang version 2.4.3, *OpenConfig Interfaces Aggregated Module*

openconfig-if-ethernet.yang version 2.12.1, *OpenConfig Interfaces Ethernet Module*

openconfig-if-ip.yang version 3.1.0, *OpenConfig Interfaces IP Module*

openconfig-if-ip-ext.yang version 2.3.1, *OpenConfig Interfaces IP Extensions Module*

openconfig-igmp.yang version 0.2.0, *OpenConfig IGMP Module*

openconfig-interfaces.yang version 3.0.0, *OpenConfig Interfaces Module*

openconfig-isis.yang version 0.3.2, *OpenConfig IS-IS Module*

openconfig-isis-policy.yang version 0.3.2, *OpenConfig IS-IS Policy Module*  
openconfig-isis-routing.yang version 0.3.2, *OpenConfig IS-IS Routing Module*  
openconfig-lacp.yang version 1.3.0, *OpenConfig LACP Module*  
openconfig-lldp.yang version 0.1.0, *OpenConfig LLDP Module*  
openconfig-local-routing.yang version 1.2.0, *OpenConfig Local Routing Module*  
openconfig-mpls.yang version 2.3.0, *OpenConfig MPLS Module*  
openconfig-mpls-ldp.yang version 3.0.2, *OpenConfig MPLS LDP Module*  
openconfig-mpls-rsvp.yang version 2.3.0, *OpenConfig MPLS RSVP Module*  
openconfig-mpls-te.yang version 2.3.0, *OpenConfig MPLS TE Module*  
openconfig-network-instance.yang version 1.1.0, *OpenConfig Network Instance Module*  
openconfig-network-instance-l3.yang version 0.11.1, *OpenConfig L3 Network Instance Module – static routes*  
openconfig-ospfv2.yang version 0.4.0, *OpenConfig OSPFv2 Module*  
openconfig-ospfv2-area.yang version 0.4.0, *OpenConfig OSPFv2 Area Module*  
openconfig-ospfv2-area-interface.yang version 0.4.0, *OpenConfig OSPFv2 Area Interface Module*  
openconfig-ospfv2-common.yang version 0.4.0, *OpenConfig OSPFv2 Common Module*  
openconfig-ospfv2-global.yang version 0.4.0, *OpenConfig OSPFv2 Global Module*  
openconfig-packet-match.yang version 1.0.0, *OpenConfig Packet Match Module*  
openconfig-pim.yang version 0.2.0, *OpenConfig PIM Module*  
openconfig-platform.yang version 0.15.0, *OpenConfig Platform Module*  
openconfig-platform-fan.yang version 0.1.1, *OpenConfig Platform Fan Module*  
openconfig-platform-linecard.yang version 0.1.2, *OpenConfig Platform Linecard Module*  
openconfig-platform-port.yang version 0.4.2, *OpenConfig Port Module*  
openconfig-platform-transceiver.yang version 0.9.0, *OpenConfig Transceiver Module*  
openconfig-procmon.yang version 0.4.0, *OpenConfig Process Monitoring Module*  
openconfig-relay-agent.yang version 0.1.0, *OpenConfig Relay Agent Module*  
openconfig-routing-policy.yang version 3.0.0, *OpenConfig Routing Policy Module*  
openconfig-rsvp-sr-ext.yang version 0.1.0, *OpenConfig RSVP-TE and SR Extensions Module*  
openconfig-system.yang version 0.10.1, *OpenConfig System Module*  
openconfig-system-grpc.yang version 1.0.0, *OpenConfig System gRPC Module*  
openconfig-system-logging.yang version 0.3.1, *OpenConfig System Logging Module*  
openconfig-system-terminal.yang version 0.3.0, *OpenConfig System Terminal Module*  
openconfig-telemetry.yang version 0.5.0, *OpenConfig Telemetry Module*  
openconfig-terminal-device.yang version 1.9.0, *OpenConfig Terminal Optics Device Module*  
openconfig-vlan.yang version 2.0.0, *OpenConfig VLAN Module*

# Customer document and product support



## Customer documentation

[Customer documentation welcome page](#)



## Technical support

[Product support portal](#)



## Documentation feedback

[Customer documentation feedback](#)