7705 Service Aggregation Router Gen 2

Release 25.3.R2

OAM and Diagnostics Guide

Table of contents

List of tables

List of figures

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 guide 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.

Unless otherwise indicated, the topics and commands described in this guide apply only to the 7705 SAR Gen 2 platforms listed in Platforms and terminology.

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

> **Note:** Unless otherwise indicated, CLI commands, contexts, and configuration examples in this guide apply for both the classic CLI and the MD-CLI.

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

- *7705 SAR Gen 2 Classic CLI Command Reference Guide*
- *7705 SAR Gen 2 Clear, Monitor, Show, Tools CLI Command Reference Guide* (for both the MD-CLI and classic CLI)
- *7705 SAR Gen 2 MD-CLI Command Reference Guide*

> **Note:** This guide generically covers Release 25.*x*.R*x* content and may contain some content that will be released in later maintenance loads. See the *SR OS R25.x.Rx Software Release Notes*, part number 3HE 21562 000*x* TQZZA, for information about features supported in each load of the Release 25.*x*.R*x* software. For a list of features and CLI commands that are present in SR OS but not supported on the 7705 SAR Gen 2 platforms, see "SR OS Features not Supported on SAR Gen 2" in the *SR OS R25.x.Rx Software Release Notes*.

1.2 Platforms and terminology

> **Note:**
> Unless explicitly noted otherwise, this guide uses the terminology defined in the following table to collectively designate the specified platforms.

Table 1: Platforms and terminology

Platform	Collective platform designation
7705 SAR-1	7705 SAR Gen 2

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

© 2025 Nokia.

Use subject to Terms available at: www.nokia.com/terms.
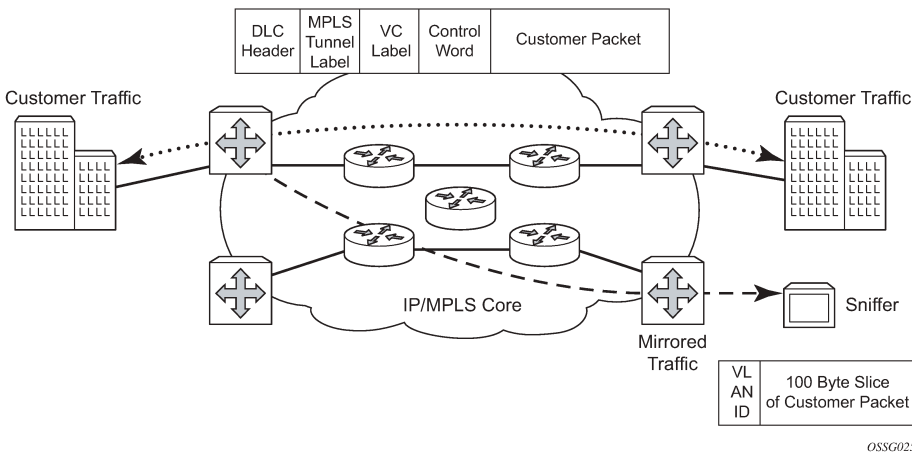
2.1 Mirror implementation

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

Enhanced subscriber management associates subscriber hosts with queuing and filtering resources in a shared SAP environment. Mirroring used in subscriber aggregation networks for 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 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 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

The user can configure multiple mirror source services, each asking for the same packets. For example, the user can configure 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 for access ports, from lowest to highest priority, is the following:

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

For example, when mirroring is enabled on a port for both filter and SAP, packets that match the 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 for network ports, from lowest to highest priority, is the following:

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 multiport 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** or **debug**>**mirror-source entries** entries. Some examples of outputs include **mirror-dest**>**sap** and **mirror-dest**>**spoke-sdp**.

- **configure**>**mirror-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.

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:

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

Mirror destination per-flow hashing support

By default, when mirroring to a SAP of type LAG or SDP with a LAG interface, the system only selects one of the port members to forward the mirrored packet.

It is possible to select per-flow hashing on the mirror destination to allow hashing of flows to all port members in a LAG.

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 7705 SAR Gen 2 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, 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
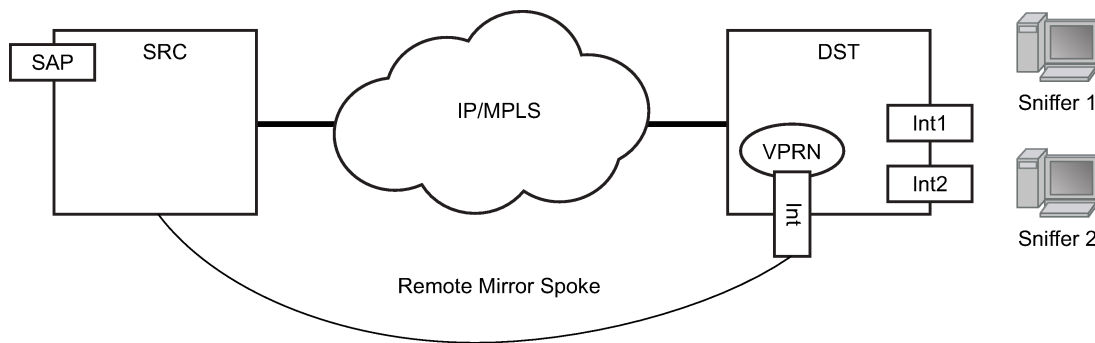
- **local mirroring termination**

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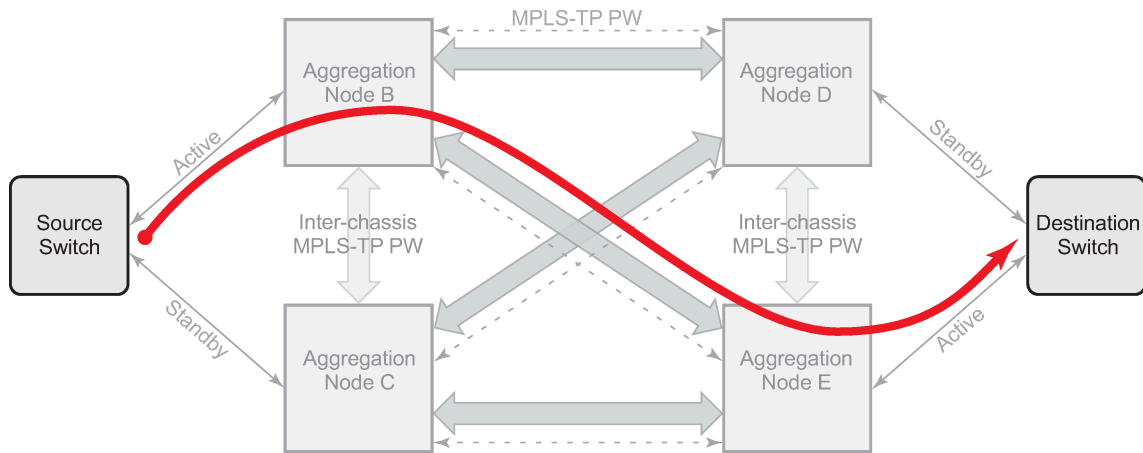
Note that mirroring traffic is usually unidirectional, flowing from "source" nodes (B or C) to "destination" nodes (D or E). However, in case of MPLS-TP, the control channel status packets may flow in the reverse direction.

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
                    saii-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
                    saii-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
    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
                        saii-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
            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
```

2.1.3.4 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 conserves mirroring resources by limiting the size of the stream of packet through the router and 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.

  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.

2.1.3.4.1 Mirror sampling rate

SR OS supports the following 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.

2.1.3.5 Mirror destination per-flow hashing support

By default, when the system mirrors to a SAP of type LAG or SDP with a LAG interface, the system only forwards the mirror packet to one port member.

A user can select per-flow hashing on the mirror destination to allow hashing of flows to all port members in a LAG.

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 Pseudowire redundant mirror services

This section describes the implementation and configuration of redundant Mirror 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 service or a mirror service where the customer has a redundant service but the mirror 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 service terminates.

Figure 4: State engine for redundant service to a redundant mirror service
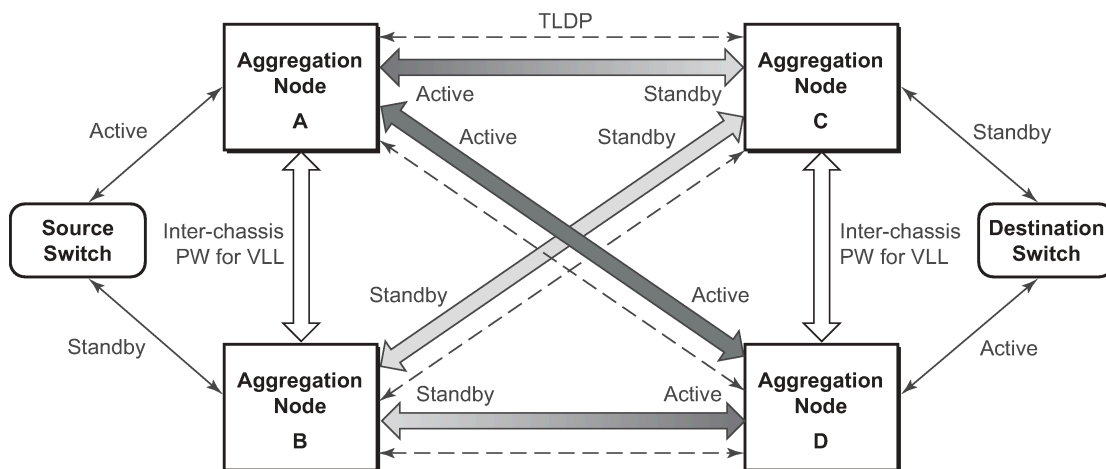
It is important to note that to provide protection if the active SDP between node A and D fails and the need to limit the number of lost data, the ICB between node A and B must be supported. As a result, when the SDP connecting nodes A and D fails the data on its way from the source switch to node A 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 service redundancy.

Figure 5: State engine for redundant service to a non-redundant mirror service
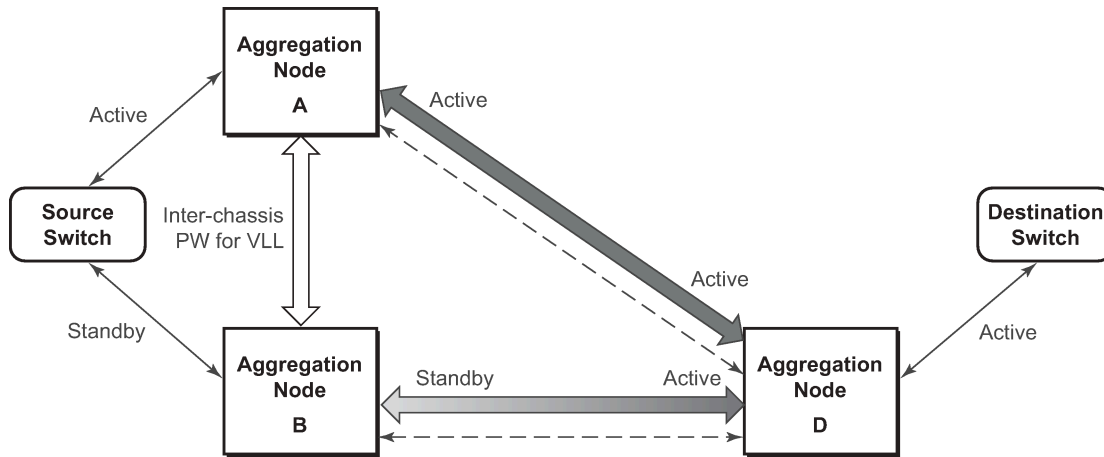
The notable difference with scenarios standard pseudo wire redundancy scenarios is that provided the customer service is redundant on nodes A and B (Figure 4: State engine for redundant service to a redundant mirror service and Figure 5: State engine for redundant service to a non-redundant mirror service) 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 4: 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 6: State engine for a non-redundant service to a redundant mirror service
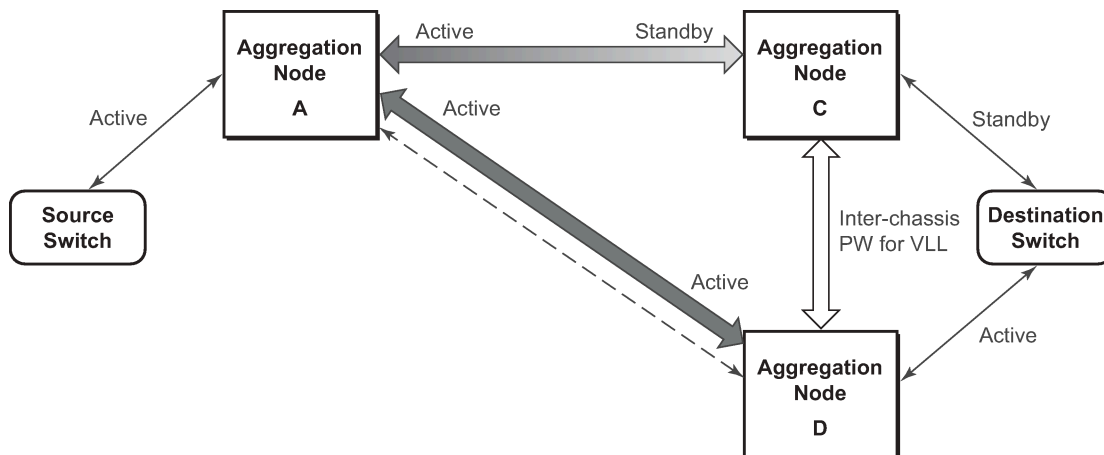
In the case where a non-redundant service is being mirrored to a redundant mirror service (Figure 6: State engine for a non-redundant service to a redundant mirror service) the source aggregation node (A) can only maintain a pseudo wire to the active destination aggregation node (D). Should the link between aggregation node D and the destination switch fail then the pseudo wire must switch to the new active aggregation node (C).

2.2.1 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.

2.3 Configuration process overview

Mirroring can be performed based on the following criteria:

- Port
- SAP
- IP filter
- Ingress label

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 7: 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 7: Local mirroring example
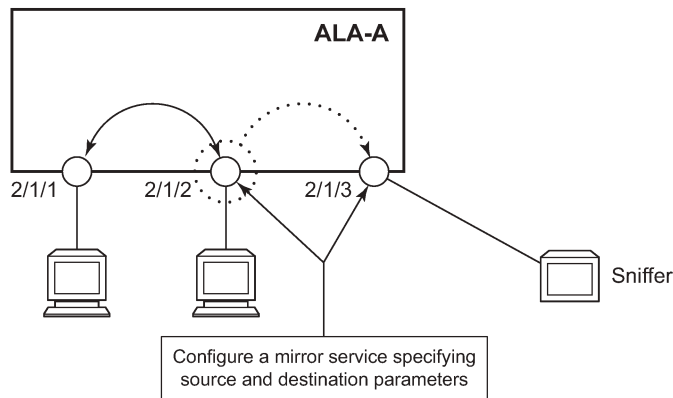
Figure 8: Remote mirroring example shows a remote mirror service configured as ALA B as the mirror source and ALA A as the mirror destination. Mirrored traffic ingressing and egressing port 5/2/1 (the source) on ALA B is handled the following ways:

- 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 8: Remote mirroring example
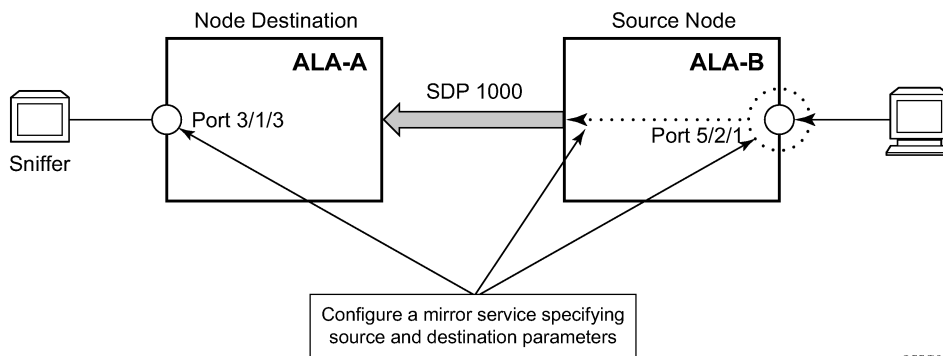
Figure 9: Mirror configuration and implementation flow shows the process to provision basic mirroring parameters.

Figure 9: Mirror configuration and implementation flow
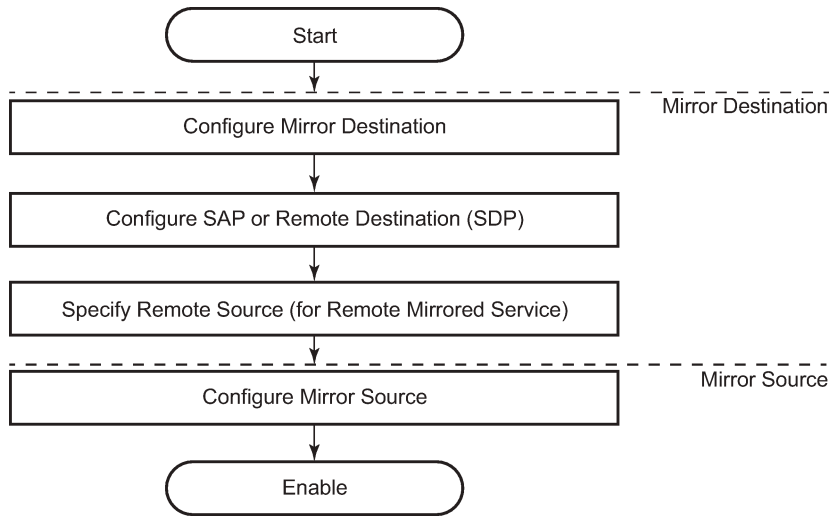
2.4 Configuration notes

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**) criteria configurations are not preserved in a configuration save (**admin save**). Debug mirror source configuration can be saved using **admin**>**debug-save**.

- 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.

2.5 Configuring service mirroring with CLI

This section provides information about service mirroring.

2.5.1 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.

2.5.1.1 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

2.5.2 Basic mirroring configuration

Mirror destination parameters must include:

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

- a mirror destination SAP or SDP

Mirror source parameters must include:

- 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 destination are on the same device.

Example

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

The following examples shows a mirror source configuration.

Example

```
*A:ALA-A>debug>mirror-source# show debug mirror
----------------------------------------------
debug
    mirror-source 103

        port 1/1/24 egress ingress
    no shutdown
    exit
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.

Example

```
*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#
```

2.5.2.1 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.

2.5.2.1.1 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 2: Port command syntax

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

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.

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

Table 3: Mirror source port requirements

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

Port type	Port mode	Port encapsulation type
Ethernet		
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
```

2.5.2.1.2 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
```

2.5.2.1.3 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.

2.5.2.1.4 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
```

2.5.3 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.

2.5.3.1 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.

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
-----------------------------------------------
```

2.5.3.2 Configuring SDPs for mirrors

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 *7705 SAR Gen 2 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.

2.5.3.2.1 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.

2.5.3.2.2 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#
```

2.5.3.3 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 10: 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 10: Remote mirrored service tasks
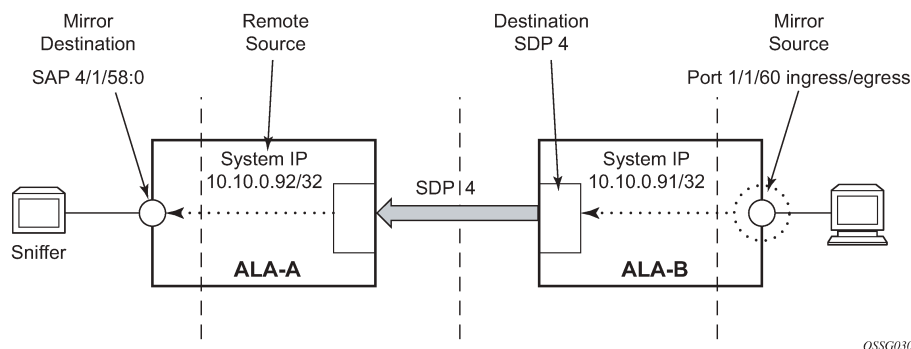
The following example shows the CLI output showing the configuration of remote mirrored service 1216. The traffic ingressing and egressing port 1/1/60 on 10.10.0.92 (ALA-B) is mirrored to the destination SAP 1/1/58:0 on ALA-A.

```
*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#
```

2.5.3.4 Pseudowire redundancy for mirror services configuration example

A configuration based on Figure 11: State engine for redundant service to a redundant mirror service is described in this section.

Figure 11: State engine for redundant service to a redundant mirror service
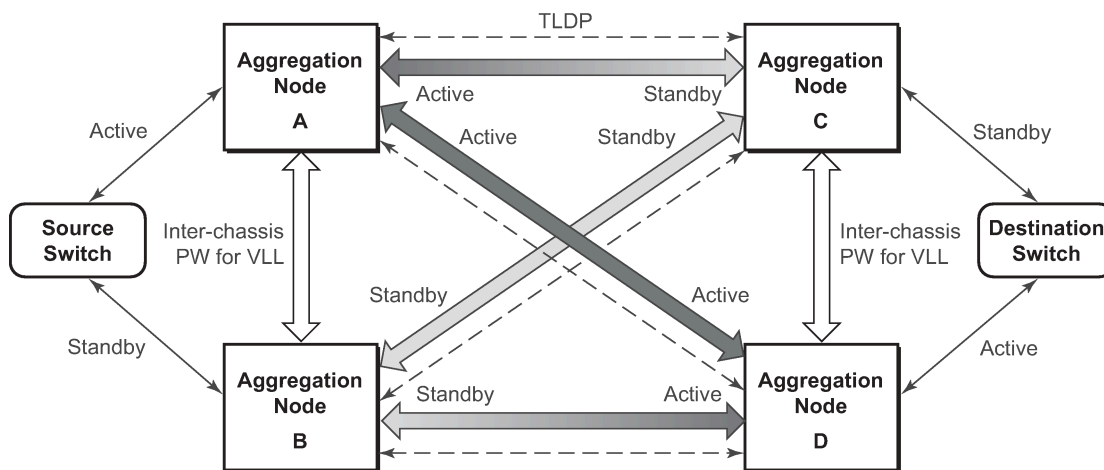
The mirror traffic needs to be forwarded from configured debug mirror-source together with mirror-dest/remote-source (ICB or non-ICB) to either SAP endpoint or SDP endpoint.

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
```

2.6 Service management tasks

This section describes service management tasks related to service mirroring.

2.6.1 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#
```

2.6.2 Deleting a local mirrored service

Existing mirroring parameters can be deleted in the CLI. A **shutdown** command must be issued at the 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.

Example: Command usage 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
```

2.6.3 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#
```

2.6.4 Deleting a remote mirrored service

Existing mirroring parameters can be deleted by using the CLI. A shut down must be issued on a service level to delete the service. It is necessary to shut down and remove SAP or SDP references to delete a remote mirrored service.

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

Example: Deleting a remote mirrored service

```
*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 preceding 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 following info command output.

Example: Mirror 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
```

Example: Debug mirror output

In the following example, 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
```

3 OAM fault and performance tools and protocols

This chapter provides information about the OAM fault and performance tools and protocols.

3.1 OAM overview

Delivery of services requires that 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 supported, 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, SR policies, SDPs, services, and VPLS MACs within a service.

3.1.1 LSP diagnostics for LDP, RSVP, and BGP labeled routes: LSP ping and LSP trace

The router LSP diagnostics include implementations of LSP ping and LSP trace based on RFC 8029, *Detecting Multiprotocol 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 labeled route, LSP ping verifies whether the packet reaches the egress label edge router (LER), while for LSP trace, the packet is sent to the control plane of each transit Label Switching Router (LSR) that performs various checks to see if it is intended to be 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 Multiprotocol 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 also 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 that follows.

3.1.1.1 LSP ping/trace for an LSP using a BGP IPv4 or IPv6 labeled route

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

The following figure shows the structure of the TLV.

Figure 12: Target FEC stack TLV for a BGP labeled IPv4 and IPv6 prefixes

```
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
┌───────────────────────────────────────────────────────────┐
│                                                             │
│              IPv4 Prefix (4 Octets) /                       │
│              IPv6 Prefix (16 Octets)                        │
│                                                             │
├───────────────────┬─────────────────────────────────────┤
│   Prefix Length   │              Must Be Zero             │
└───────────────────┴─────────────────────────────────────┘
                                                        sw3092
```
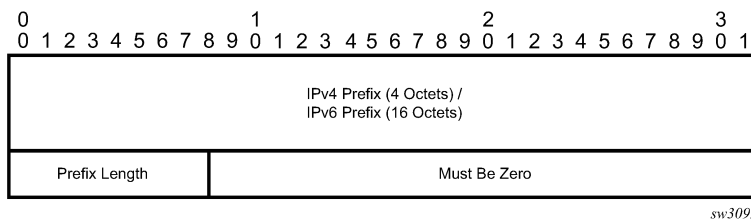
The user issues an LSP ping using the following command and specifies a bgp-label type of prefix:

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 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 can issue an 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 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.

1. The next hop of a BGP labeled route for an IPv4 /32 or IPv6 /128 prefix can be resolved to either an IPv4 or IPv6 transport tunnel. The sender node encapsulates the packet of the Echo Request message with a label stack that 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 that acts as an LSR for the outer transport LSP, and the node does not have context for the BGP label prefix, then it validates the outer label in the stack. If the validation is successful, it replies the same way that it does when it receives an Echo Request message for an LDP FEC that is stitched to a BGP IPv4 labeled route. That is, it replies with return code 8 "Label switched at stack-depth <RSC>".

2. An LSR node that is the next hop for the BGP label prefix and the LER node that originated the BGP label prefix have full context for the BGP IPv4 or IPv6 target FEC stack and can perform full validation of it.

3. If a BGP IPv4 labeled 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 behavior is

the same as an LDP FEC that is stitched to a BGP IPv4 labeled route when the Echo Request message reaches the egress LER for the BGP prefix.

Note: Only BGP label IPv4 /32 prefixes and BGP IPv6 /128 prefixes are supported because only these prefixes 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 labeled route resolves to an IPv6 next hop and uses an IPv6 transport tunnel, any LSR or LER node that 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 labeled route when resolved to an IPv4 or an IPv4-mapped IPv6 next hop, which itself is resolved to an IPv4 transport tunnel.

3.1.1.2 LSP ping and LSP trace over unnumbered IP interface

LSP ping for Point-to-Point (P2P) and Point-to-Multipoint (P2MP) LSPs can 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 an 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 defined in RFC 8029. Both values are taken from the TE database.

In an 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 as defined 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.

3.1.1.3 ECMP considerations for LSP ping and LSP trace

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

When BGP labeled route is resolved to an LDP FEC (of the BGP next-hop of the BGP labeled 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 labeled 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 labeled route is equal to the number of BGP next-hops.

- ECMP for a BGP labeled 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 labeled 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 labeled route. Furthermore, it returns exactly the LDP next-hop the datapath 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 labeled 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, the sender node does not include multipath information in the DSMAP TLV in the echo request message (multipath type=0). In this case, the responder node replies with a DSMAP TLV for each outgoing interface, which is part of the ECMP next-hop set for the FEC.

  > **Note:** The sender node selects the first DSMAP 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, the sender node does not include the DSMAP 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**}, the sender node does not include the DSMAP 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, 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 DSMAP TLV. The CPM code at each LSR in the path of the target FEC runs the same hash routine as the datapath 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 **lsr-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 **lsr-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 **lsr-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 preceding 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:** If the user enabled the **system-ip-load-balancing hash** option (**config**>**system**>**system-ip-load-balancing**), 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 in the preceding sections.

- 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 ddmap**) or within the specific **ldp-treetrace** or **lsp-trace** test (**downstream-map-tlv ddmap option**).

3.1.1.4 LSP ping for RSVP P2MP LSP (P2MP)

The P2MP LSP ping implementation complies with 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 datapath over all branches of the P2MP LSP instance. By default, all egress LER nodes that 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 five addresses can be specified in a single execution of the **p2mp-lsp-ping** command. If all five 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. 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 multiple times 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; no traps are issued 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 **ttl** 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:** 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 does 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 does not cause an error in processing the downstream mapping TLV.

If the **ping** expires at an LSR node that is performing a remerge or crossover operation in the datapath 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 **p2mp-lsp-ping** 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 displays 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 has 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 *7705 SAR Gen 2 MPLS Guide*.

3.1.1.5 LSP trace for RSVP P2MP LSP

The P2MP LSP trace is in accordance with RFC 6425. Generate an LSP trace using 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 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 the maximum number of probes, 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 that replied. This process 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 interfaces of the FEC.

```
     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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |              MTU                       | Address Type   |    DS Flags
      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |              Downstream Address (4 or 16 octets)
      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |           Downstream Interface Address (4 or 16 octets)
      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |  Return Code  | Return Subcode    |        Subtlv Length
      |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      .
                       .
      .                      List of SubTLVs

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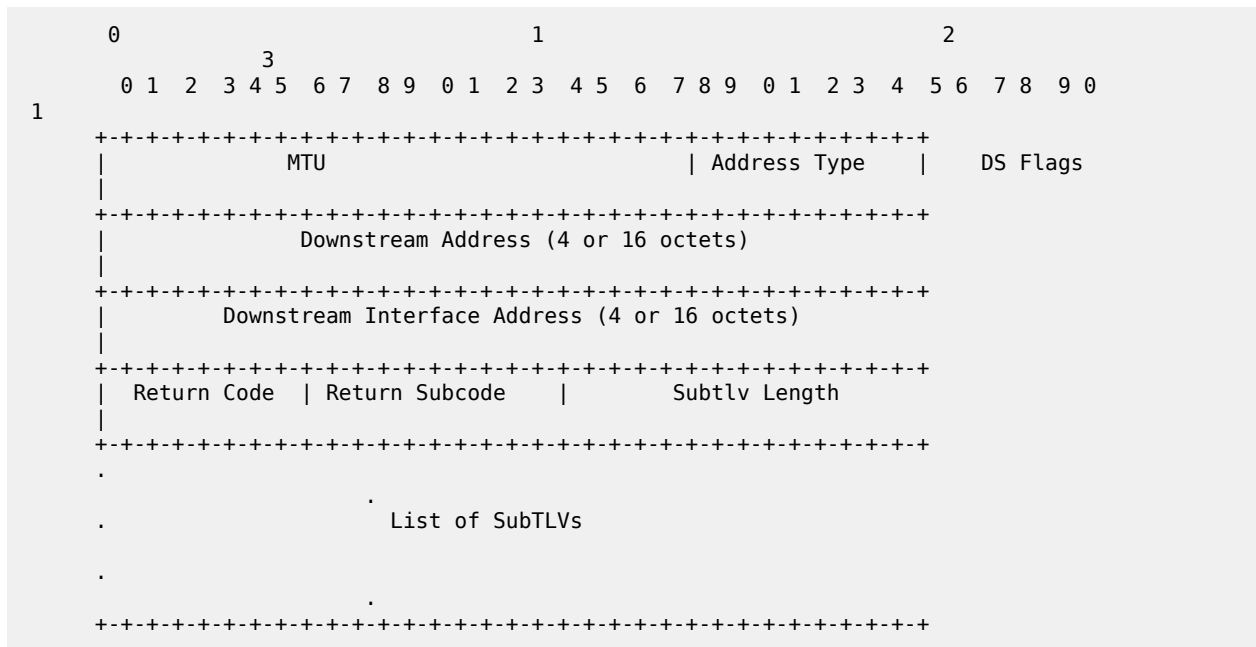
The Downstream Detailed Mapping TLV format is derived from the Downstream Mapping (DSMAP) TLV format. The key change is that variable length and optional fields have been converted into sub-TLVs. The fields have the same use and meaning as in RFC 8029.

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.

As well, 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 that 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.

3.1.1.5.1 LSP trace behavior when S2L path traverses a remerge node

When a 7705 SAR Gen 2 LSR performs a remerge 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 that remerge 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)

    A
  /   \
B -- C
|
D
| \
F   E
```
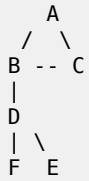
- **tracing S2L1 when ILM on interface C-B blocked at node B**

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

  For TTL=2, assume A receives 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 receives 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 remerge 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 the potential of a remerge 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 4: Echo reply messages and ingress LER behavior.

Table 4: 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 remerge but would continue trace on first 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 that 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 remerge 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.

3.1.1.6 Downstream Detailed Mapping (DDMAP) TLV

The Downstream Detailed Mapping (DDMAP) TLV provides the same features as the DSMAP TLV, with the enhancement to trace the details of LSP stitching and LSP hierarchy. The latter is achieved using a sub-TLV of the DDMAP TLV called the FEC stack change sub-TLV. Figure 13: DDMAP TLV shows the structures of these two objects as defined in RFC 6424.

Figure 13: DDMAP TLV


```
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
```

MTU	Address Type	DS Flags
Downstream Address (4 or 16 octets)		
Downstream Interface Address (4 or 16 octets)		
Return Code	Return SubCode	Sub-tlv Length
List of Sub TLVs		

sw0862
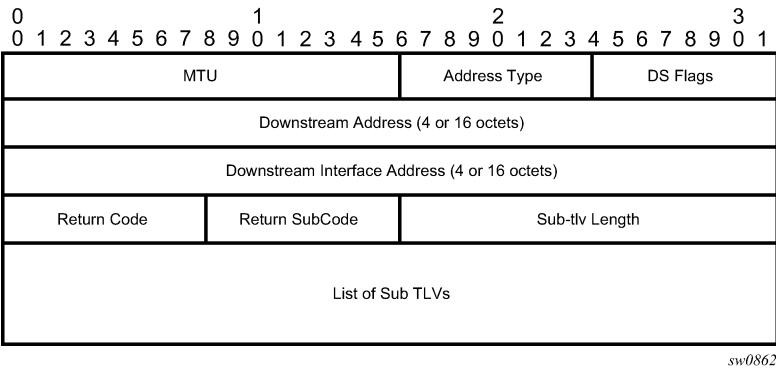
The DDMAP TLV format is derived from the DSMAP TLV format. The key change is that variable length and optional fields have been converted into sub-TLVs. The fields have the same use and meaning as in RFC 8029 as shown in Figure 14: FEC stack change sub-TLV.

Figure 14: FEC stack change sub-TLV


```
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
```

Operation Type	Address Type	FEC-tlv Length	Reserved
Remote Peer Address (0, 4 or 16 octets)			
FEC TLV			

sw0860
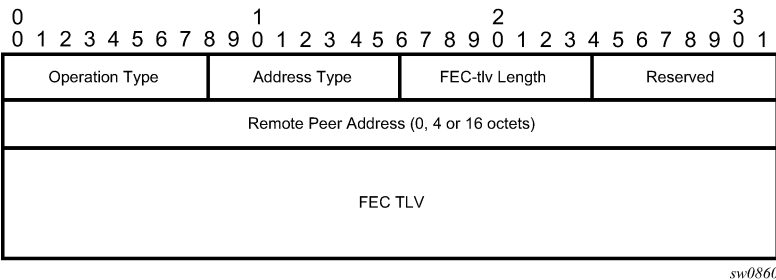
The operation type specifies the action associated with the FEC stack change. The following operation types are defined.

```
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** | **ddmap**}

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 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 labeled 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 DSMAP 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** | **ddmap** | **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** | **ddmap** | **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 preceding rules is that a change to the value of the **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 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 that 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 an 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 **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 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.

3.1.1.7 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 labeled 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 labeled 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 labeled 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 labeled route or of a BGP IPv6 labeled 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 labeled route (with an IPv6 next-hop) or a BGP IPv6 labeled 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 labeled route (with an IPv4 or an IPv4-mapped IPv6 next-hop) recursively resolved to a BGP IPv4 labeled 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 correctly check a target FEC that is stitched to another FEC (stitching FEC) of the same or a different type, or that 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 operations 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 changes 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 that are a superset of the rules described in Section 4 of RFC 6424 to allow greater scope of interoperability with other vendor implementations.

3.1.1.7.1 Responder node procedures

This section describes responder-node behaviors.

1. As a responder node, the node 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"

2. When the responder node inserts a global return code of 3, it does not include a DDMAP TLV.

3. 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 does the following:
   - On a success response, includes a return code of 15 in the DDMAP TLV for each downstream that has an FEC stack change TLV.
   - On a success response, includes 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, includes an appropriate error return code in the DDMAP TLV for each downstream.

4. A tunneling node indicates that it is pushing an FEC (the tunneling FEC) on top of the target FEC stack TLV by including an FEC stack change sub-TLV in the DDMAP TLV with an 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.

5. A node that is stitching an 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 therefore 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.

6. If the responder node is the egress for one or more FECs in the target FEC stack, 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 resends the echo request message with the same TTL value. 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 again follows steps 1 or 2.

3.1.1.7.2 Sender node procedures

This section describes sender-node behaviors.

1. 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. This reflects 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. That is, one FEC is popped at most and one or more FECs are pushed as indicated.

2. If the Echo Reply message contains the return code 3 "Replying router is an egress for the FEC at stack-depth <RSC>":

    a. If the value for the label stack depth specified in the Return Sub-Code (RSC) field is the same as the depth of the current target FEC stack TLV, the sender node considers the trace operation complete and terminates it. A responder node causes this case to occur as per step 6 of Responder node procedures.

    b. 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. This step continues iteratively until the value for the label stack depth specified in the RSC field is the same as the depth of the current target FEC stack TLV, and in which case, the preceding step is performed. A responder node causes this case to occur as per step 6 of the Responder node procedures.

    c. If a DDMAP TLV with or without an FEC stack change sub-TLV is included, the sender node must ignore it and processing is performed as per the first or second preceding steps. A responder node does not cause this case to occur, but a third-party implementation may do so.

3. As a sender node, it 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 correctly process the FEC stack change TLV as per step 1.

4. 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 therefore, the responder node, which is the egress node,

still 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.

3.1.2 OAM support in Segment Routing with MPLS data plane

MPLS OAM supports Segment Routing extensions to **lsp-ping** and **lsp-trace** as defined 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 midpoint 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

3.1.2.1 OAM support in IPv4 or IPv6 SR policies with MPLS data plane

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 the **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** | **ddmap** | **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 datapath.

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

3.1.2.1.1 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 (for SIDs of the SR policy except the top SID) or if there is no explicit FEC associated with the label (for 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.

- For **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.

- For **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. Looks 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, continues processing of the NIL FEC.

  3. Otherwise, looks up the label of the NIL FEC in the Label Manager.

  4. If a match exists, processes the FEC as per the POP or SWAP operation provided by the lookup and following the NIL FEC procedures in RFC 8029.

  5. Otherwise, fails 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, 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, 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.

3.1.2.1.2 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 datapath 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.

3.1.2.2 SR extensions for lsp-ping and lsp-trace CLI commands

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

An SR shortest path tunnel for SR IS-IS or SR-OSPF 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.

The following figure shows the format of the IPv4 IGP-prefix segment ID.

Figure 15: IPv4 IGP-prefix segment ID
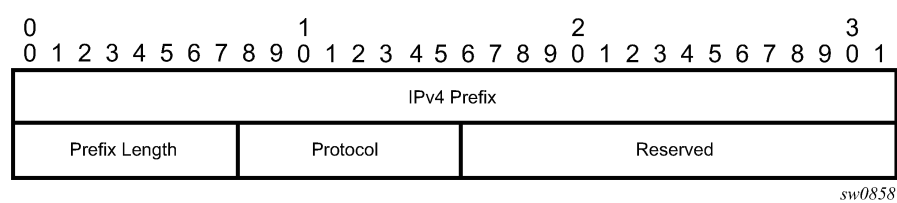
The IPv4 IGP-prefix segment ID consists of the following fields:

*   **IPv4 Prefix**

    This field is the IPv4 prefix to which the segment ID is assigned. For anycast segment ID, this field is the IPv4 anycast address. If the prefix is shorter than 32 bits, trailing bits must be set to zero.

*   **Prefix Length**

    This field is one octet and is 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 set to 2 if the IGP protocol is IS-IS.

The following figure shows the format for the IPv6 IGP-prefix segment ID.

Figure 16: IPv6 IGP-prefix segment ID
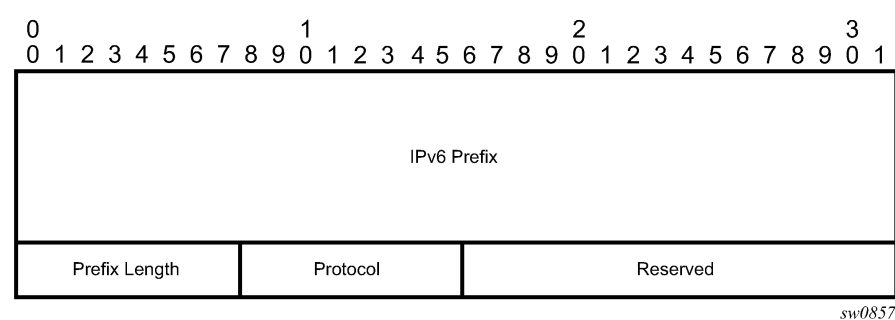
In this format, the fields are as follows:

*   **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**

    This field is one octet and provides 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 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 an 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 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 described previously in this section.

The following figure shows the format of the IGP-Adjacency segment ID.

Figure 17: IGP-Adjacency segment ID


```
 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
```

Adj. Type	Protocol	Reserved

Local Interface ID (4 or 16 octets)

Remote Interface ID (4 or 16 octets)

Advertising Node Identifier (4 or 6 octets)

Receiving Node Identifier (4 or 6 octets)
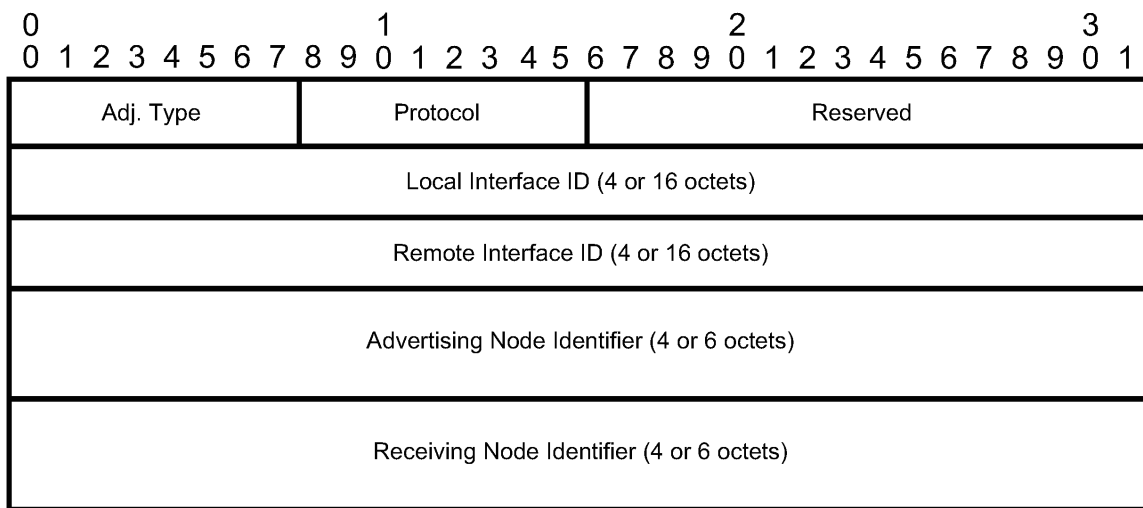
sw0859

The IGP-Adjacency segment ID consists of the following fields:

- **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 the 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, this field uses the 32-bit link identifier defined in RFC 4203 and RFC 5307. 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.

- **Remote Interface ID**

  This field is an identifier that is assigned by the remote LSR for a link on which the adjacency segment ID is bound. This field is set to the remote (downstream neighbor) link address (IPv4 or IPv6). If unnumbered, the field uses the 32-bit link identifier defined in RFC 4203 and RFC 5307. 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, the 32 right-most bits represent the OSPF router ID. If the Protocol field is set to 2, this field is 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, the 32 right-most bits represent the OSPF router ID. If the Protocol field is set to 2, this field is the 48-bit IS-IS system ID.

Both **lsp-ping** and **lsp-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

3.1.2.3 Operations on SR IS-IS or SR-OSPF tunnels

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

- 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 IS-IS 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 defined in RFC 8029.

- A node SID label that is popped at an LSR results in return code 3, "Replying router is an egress for the FEC at stack-depth <RSC>".

- The **lsp-trace** command is supported with the inclusion of the DSMAP TLV, the DDMAP TLV, or none values (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.

The following figure shows an example topology for an **lsp-ping** and **lsp-trace** command for SR IS-IS node SID tunnel.

Figure 18: Testing MPLS OAM with SR tunnels
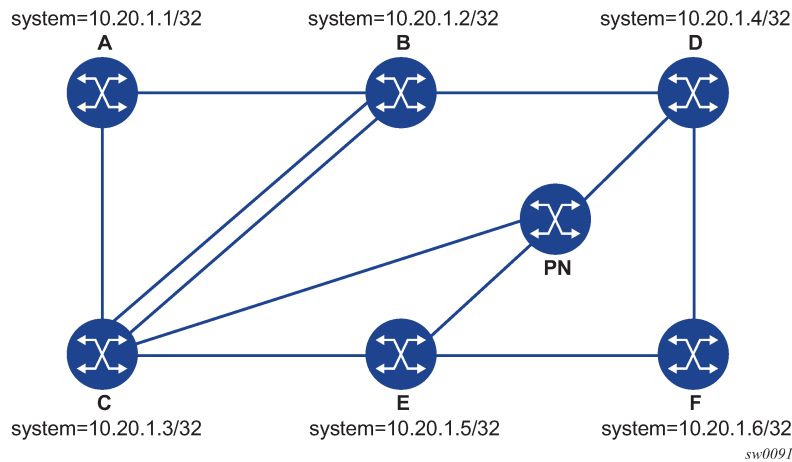
The following examples use the sample topology shown in the preceding figure.

Example: lsp-ping command on DUT-A for target node SID of DUT-F (DDMAP TLV)

```
*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
```

Example: lsp-trace command on DUT-A for a 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
```

Example: lsp-trace command 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 ddmap 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
```

3.1.2.4 Operations on SR-TE LSP

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

- 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 an 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 return code 3, "Replying router is an egress for the FEC at stack-depth <RSC>".

  An adjacency SID label popped at an LSR results in return code 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>" as defined in 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>" as defined in 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 values (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 and the protocol that 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 header is higher than the depth of the Target FEC stack TLV.

- TTL values can be changed.

  The **ttl** value in the **lsp-ping** command 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 the **lsp-trace** command can be set to values other than default. The minimum TTL value can, however, 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 because the responder node receives 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 figure shows an example topology for the **lsp-ping** and **lsp-trace** commands for SR-TE LSPs.

Figure 19: Testing MPLS OAM with SR-TE LSP
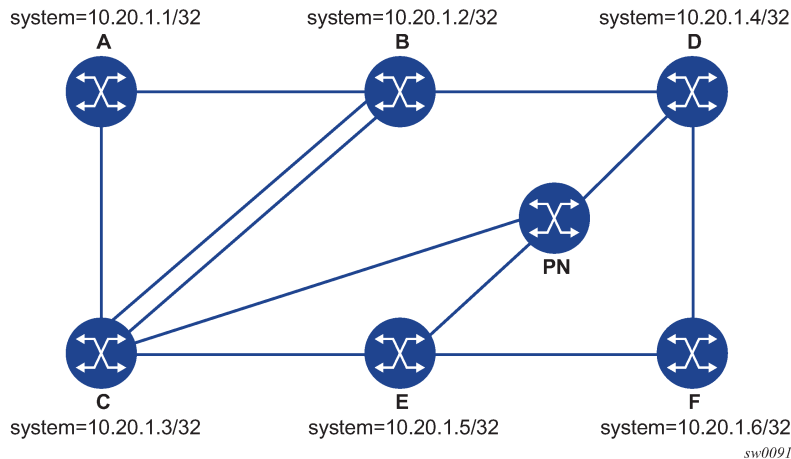
The following are examples show outputs for the **lsp-ping** and **lsp-trace** commands for SR-TE LSPs, based on the sample topology shown in the preceding figure. Example 1 uses a path with strict hops, each corresponding to an adjacency SID, while Example 2 uses a path with loose hops, each corresponding to a node SID.

Example: 1

The following output is an example of **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 "srteABCEDF" detail
LSP-PING srteABCEDF: 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 srteABCEDF 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 "srteABCEDF" downstream-map-tlv ddmap detail
lsp-trace to srteABCEDF: 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
```

Example: 2

The following output is an example of **lsp-ping** and **lsp-trace** on DUT-A for a 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 ddmap 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
```

3.1.2.5 Operations on an SR IS-IS tunnel stitched to an LDP FEC

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

- The **lsp-ping** tool works only when the responder node is in the same domain (SR or LDP) as the sender node.

- The **lsp-ping** 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 an SR IS-IS tunnel stitched to an LDP FEC.

Example: 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 ddmap
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=10.1
0.3.2
2  10.20.1.2  rtt=4.32ms rc=3(EgressRtr) rsc=1
*A:Dut-E#
```

Example: 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 ddmap 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
```

3.1.2.6 Operations on a BGP IPv4 LSP resolved over an SR IS-IS IPv4 tunnel, SR-OSPF IPv4 tunnel, or SR-TE IPv4 LSP

The operations of LSP ping and LSP trace of a BGP IPv4 LSP resolved over an SR IS-IS IPv4 tunnel, an SR-OSPF IPv4 tunnel, or an SR-TE IPv4 LSP are enhanced. The 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 module 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.

The following figure shows an example topology used in the subsequent BGP over SR-OSPF, BGP over SR-TE (OSPF), BGP over SR IS-IS, and BGP over SR-TE (IS-IS) examples.

Figure 20: Example topology for BGP over SR-OSPF, SR-TE (OSPF), SR IS-IS, and SR-TE (IS-IS)
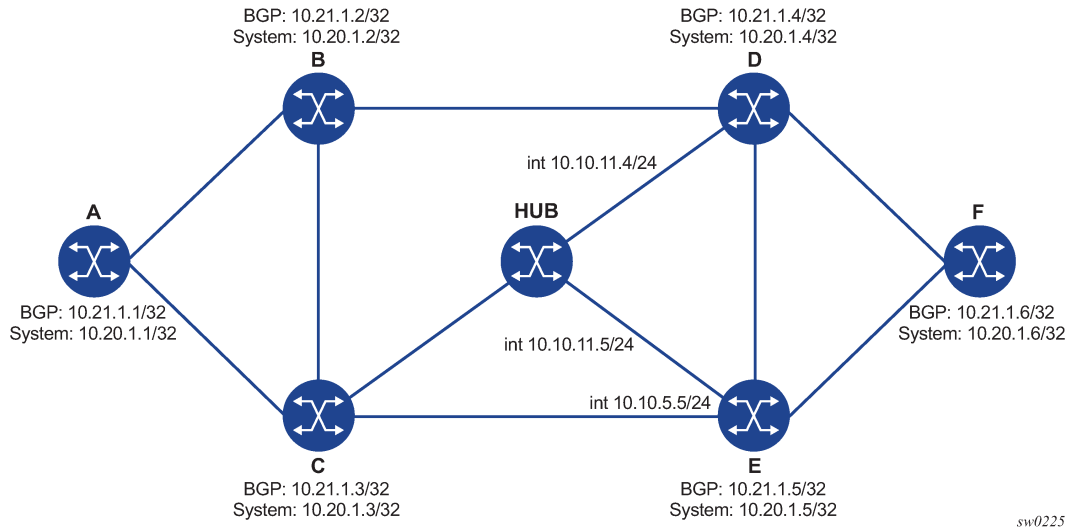
The following are examples of the **lsp-trace** command output for a hierarchical tunnel consisting of a BGP IPv4 LSP resolved over an SR IS-IS IPv4 tunnel, SR-OSPF IPv4 tunnel, or SR-TE IPv4 LSP.

Example: BGP over SR-OSPF

```
*A:Dut-A# oam lsp-trace bgp-label prefix 11.21.1.6/32 detail downstream-map-
tlv ddmap path-destination 127.1.1.
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#
```

Example: BGP over SR-TE (OSPF)

```
*A:Dut-A# oam lsp-trace bgp-label prefix 11.21.1.6/32 detail downstream-map-
tlv ddmap 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#
```

Example: BGP over SR IS-IS

```
A:Dut-A# oam lsp-trace bgp-label prefix 11.21.1.6/32 detail downstream-map-
tlv ddmap 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
```

Example: BGP over SR-TE (IS-IS)

```
*A:Dut-A# oam lsp-trace bgp-label prefix 11.21.1.6/32 detail downstream-map-
tlv ddmap 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
```

The following figure shows the topology with the addition of an eBGP peering between nodes B and C, the BGP IPv4 LSP spans the AS boundary and resolves to an SR IS-IS tunnel or an SR-TE LSP within each AS.

Figure 21: Example topology for BGP over SR IS-IS in inter-AS option C and BGP over SR-TE (IS-IS) in inter-AS option C
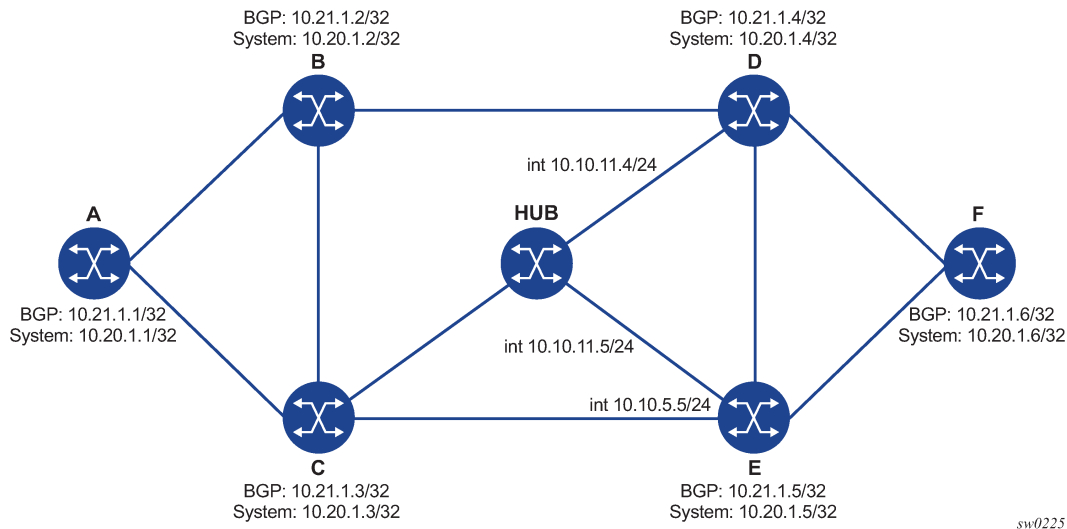
Example: BGP over SR IS-IS in inter-AS option C

```
*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 ddmap 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=10.1
0.5.5
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
```

Example: BGP over SR-TE (IS-IS) in inter-AS option C

```
*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 ddmap 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
```

3.1.2.7 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 22: 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 22: Example topology for SR-ISIS over RSVP-TE and SR-OSPF over RSVP-TE
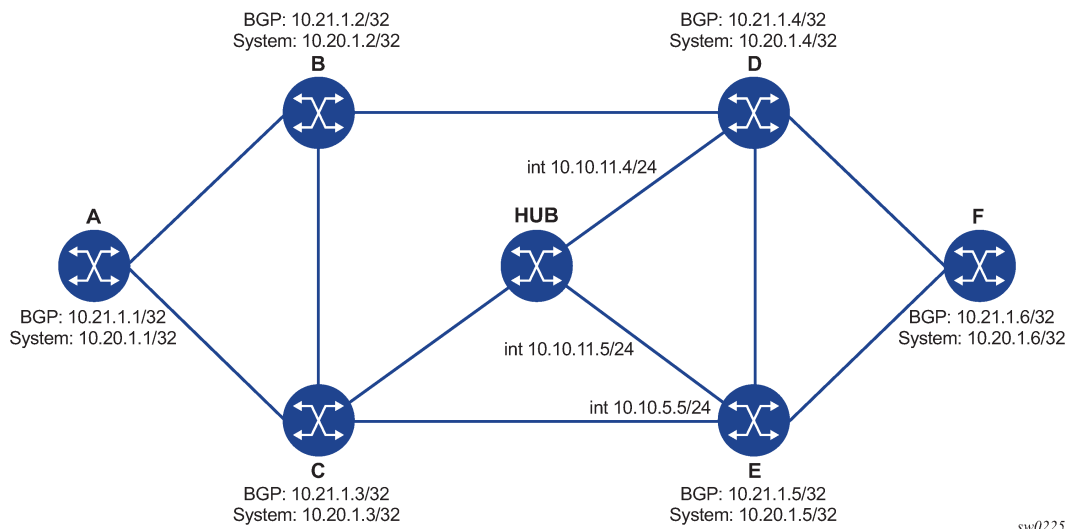
Example: SR-ISIS over RSVP-TE

```
*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#
```

Example: SR-OSPF over RSVP-TE

```
*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
```

3.1.2.8 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 example outputs show 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 23: 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 23: Example topology for LDP over SR-TE (ISIS) and LDP over SR-TE (OSPF)
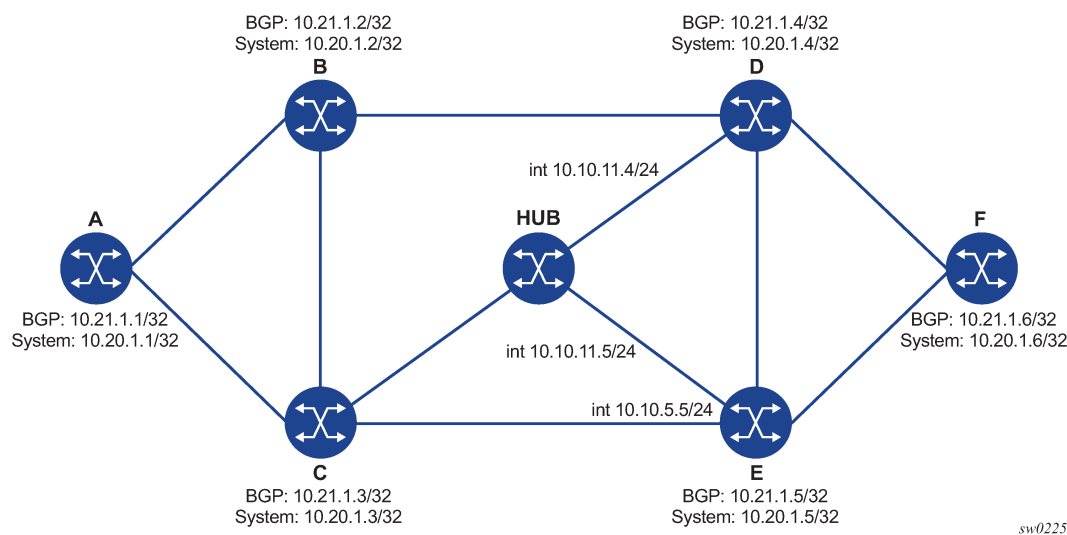
Example: LDP over SR-TE (ISIS)

```
*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
===============================================================================
```

Example: LDP over SR-TE (OSPF)

```
*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
===============================================================================
```

3.1.3 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 defined in RFC 3032. At an LSR node, including an ABR, ASBR, or datapath 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:** 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, 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 datapath in the GRT context for BGP shortcut, 6PE, and BGP labeled 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:** 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 7705 SAR Gen 2 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 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 Multipart Messages*, which defines extensions for a multipart 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 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.

- 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:** 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 times out 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.

3.1.3.1 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.

3.1.3.2 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 information:

- **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 the 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 tunnel 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 the 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 tunnel 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 labeled route). In all cases, the ICMP reply packet uses the outgoing interface address as the source address of the reply packet.

3.1.4 SDP diagnostics

The router SDP diagnostics are SDP ping and SDP MTU path discovery.

3.1.4.1 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 they follow 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

3.1.4.2 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).

3.1.5 Service diagnostics

The service diagnostics include the following:

- service ping
- IGMP snooping
- various VPLS MAC diagnostic tools
- various VLL diagnostic tools

3.1.5.1 IGMP snooping diagnostics

3.2 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

This section provides information about TWAMP Light and STAMP.

3.2.2.1 Overview

Within SR OS, the **twamp-light** container contains configuration elements for IP Performance Measurement (IP PM) tests. This is a direct correlation to the actual test PDU format, TWAMP Light or STAMP. The various IP PM features use the appropriate test PDU format. Some applications may have strict requirements dictating the test PDU format, while others are more flexible where the test PDU depends on configuration.

> **Note:** This guide uses TWAMP Light as generic terminology. The usage does not suggest the application test PDU format.

SR OS is an early adopter of IP PM under the OAM Performance Monitoring (OAM-PM) architecture. The test PDU format was derived from TWAMP Light based on RFC 5357, section "Informational Appendix I". As TWAMP Light gained community support and large deployment, it became the defacto standard for IP performance measurement. Following community acceptance, the IETF published RFC 7862, *Simple Two-way Active Measurement Protocol (STAMP)*.

In link measurement, the Session-Sender uses STAMP-formatted packets because of the requirement to support the TLV structure described in RFC 8972, *STAMP Optional Extensions*.

In link measurement on LAG, the Session-Sender uses TWAMP Light-formatted packets in accordance with the reassignment of TWAMP Light test PDU fields that were previously Must Be Zero (MBZ).

In OAM-PM, the Session-Sender allows for a choice of TWAMP Light or STAMP test PDU formats.

```
configure oam-pm session ip twamp-light session-sender-type
```

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.

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 with RFC 5357 Appendix I. As the STAMP work in the IETF continues to evolve, the backward compatibility has remained largely unchanged between the RCF 5357 appendix and the base STAMP RFC 8762 . Even with the publication of RFC 8972, *Simple Two-Way Active Measurement Protocol Optional Extension* reasonable interoperability has been maintained.

Unless specifically required, the Session-Reflector should be configured with **type stamp** to provide the highest level of interoperability for different Session-Senders.

Use the following command to configure the test packet processing behavior for the Session-Reflector.

```
configure router twamp-light reflector type
```

The Session-Reflector uses the **twamp-light** and **stamp** options to determine its processing behavior for the test packet received from the Session-Sender. The default processing behavior is that **twamp-light** performs no TLV processing, treating any non-base TWAMP Light packet as padding. If the required behavior for the Session-Reflector is to parse and process STAMP TLVs, the **stamp** option must be used. Using the **stamp** 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 test PDU-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 by default. 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.

- **STAMP**
  The STAMP test packet request and response sizes are symmetrical by default. RFC 8762 defines a structured packet that ensures this behavior. To allow for 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.

3.2.2.2 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 Session-Reflector does not match the configured prefixes, the packet is dropped. Multiple prefix entries may be configured per context on the Session-Reflector. 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** command implements 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 enable until a value in the allowable range is configured. A non-ISSU upgrade can proceed regardless of the state (enabled or disabled) of the TWAMP Light Session-Reflector. The configuration can load, but the TWAMP Light Session-Reflector remains inactive following the reload when the allowable range is not configured.
>
> 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 for that session.

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 the Z bit to indicate the format of the needed 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 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 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 OAM Performance Monitoring for more information about the integration of TWAMP Light in related applications.

3.3 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.

3.3.1 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 (multihop 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.

3.3.2 Control packet format

The BFD control packet has two sections: a mandatory section and an optional authentication section.

Figure 24: Mandatory frame format

al_0893
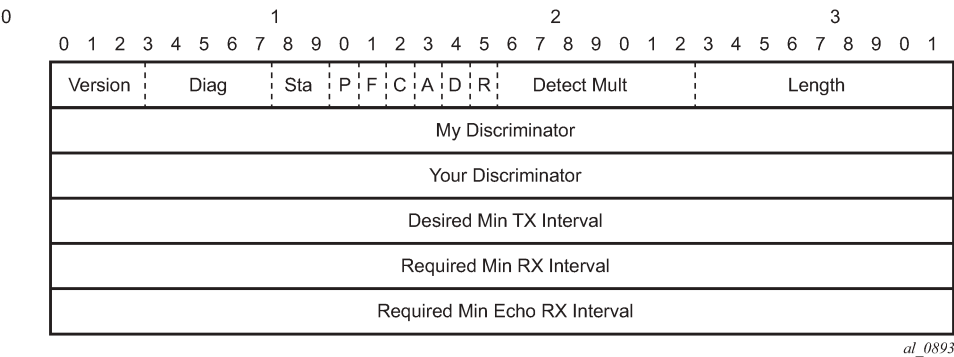
Table 5: BFD control packet field descriptions

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.

Field	Description
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.

3.3.3 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.

3.4 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.

3.4.1 BGP-LDP stitching and ASBR/ABR/datapath RR for BGP IPv4 labeled route

```
ASBR1      ASBR2
           -------- D ------- E --------
           |                   |
           |                   |
A -------- C                   F -------- B
DSLAM1     PE1                 PE2        DSLAM2
           |                   |
           |                   |
           -------- G ------- H --------
```
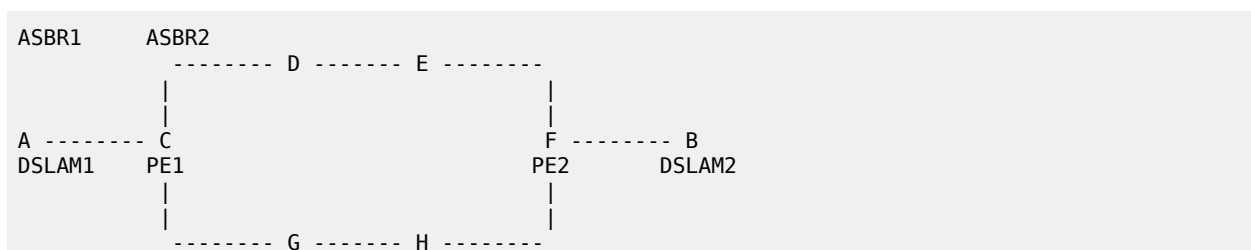
```
                      ASBR3      ASBR4

# lsp-trace ldp-bgp stitching
*A:Dut-A# oam lsp-trace prefix 10.20.1.6/32 detail downstream-map-
tlv ddmap
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 byt
e 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
```

3.4.2 VPRN inter-AS option B

```
        ASBR1      ASBR2
              -------- D ------- E --------
              |                     |
              |                     |
A -------- C                        F -------- B
CE1        PE1                      PE2        CE2
              |                     |
              |                     |
              -------- G ------- H --------
                    ASBR3     ASBR4

# 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
```

3.4.3 VPRN inter-AS option C and ASBR/ABR/datapath RR for BGP IPv4 labeled route
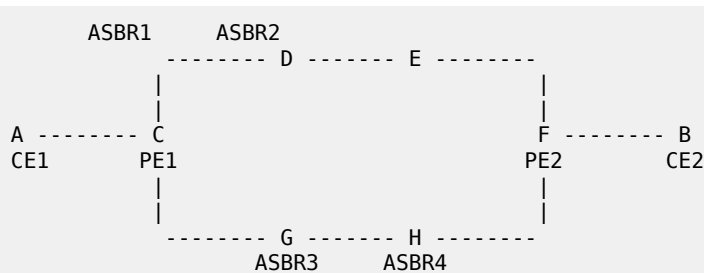
```
        ASBR1      ASBR2
              -------- D ------- E --------
              |                     |
              |                     |
A -------- C                        F -------- B
CE1        PE1                      PE2        CE2
              |                     |
              |                     |
              -------- G ------- H --------
                    ASBR3      ASBR4


# 12.0R4 default behavior

*A:Dut-B# traceroute 16.1.1.1 source 26.1.1.2 detail no-dns
wait 100 traceroute to 16.1.1.1 from 26.1.1.2, 30 hops max, 40 byte packets
  1   1  26.1.1.1  1.90 ms
  1   2  26.1.1.1  1.81 ms
  1   3  26.1.1.1  2.01 ms
  2   1  16.1.1.1  6.11 ms
  2   2  16.1.1.1  8.35 ms
  2   3  16.1.1.1  5.33 ms


*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  26.1.1.1  5.03 ms
  1   2  26.1.1.1  4.60 ms
  1   3  26.1.1.1  4.60 ms
  2   1  26.1.1.2  6.54 ms
  2   2  26.1.1.2  5.99 ms
  2   3  26.1.1.2  5.74 ms


# With ttl-propagate vprn-transit all and icmp-tunneling

*A:Dut-B# traceroute 16.1.1.1 source 26.1.1.2 detail no-dns
```

```
wait 100 traceroute to 16.1.1.1 from 26.1.1.2, 30 hops max, 40 byte packets
  1   1  26.1.1.1  2.05 ms
  1   2  26.1.1.1  1.87 ms
  1   3  26.1.1.1  1.85 ms
  2   1  10.10.4.4  8.42 ms
            returned MPLS Label Stack Object
               entry  1:  MPLS Label =  262143, Exp = 0, TTL =   1, S = 0
               entry  2:  MPLS Label =  262137, Exp = 0, TTL =   1, S = 0
               entry  3:  MPLS Label =  262142, Exp = 0, TTL =   1, S = 1
  2   2  10.10.4.4  5.85 ms
            returned MPLS Label Stack Object
               entry  1:  MPLS Label =  262143, Exp = 0, TTL =   1, S = 0
               entry  2:  MPLS Label =  262137, Exp = 0, TTL =   1, S = 0
               entry  3:  MPLS Label =  262142, Exp = 0, TTL =   1, S = 1
  2   3  10.10.4.4  5.75 ms
            returned MPLS Label Stack Object
               entry  1:  MPLS Label =  262143, Exp = 0, TTL =   1, S = 0
               entry  2:  MPLS Label =  262137, Exp = 0, TTL =   1, S = 0
               entry  3:  MPLS Label =  262142, Exp = 0, TTL =   1, S = 1
  3   1  10.10.1.2  5.54 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
  3   2  10.10.1.2  7.89 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
  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
```
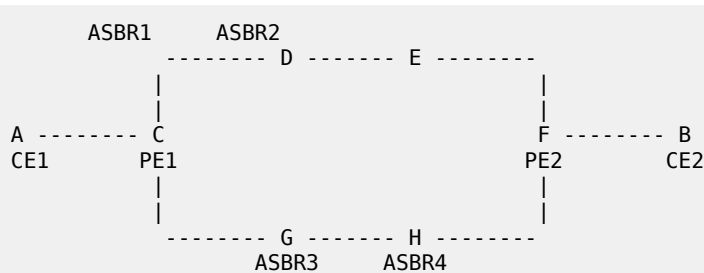
```
            entry  1:  MPLS Label =  262138, Exp = 7, TTL =   1, S = 0
            entry  2:  MPLS Label =  262142, Exp = 7, TTL =   2, S = 1
 3   1  26.1.1.1  4.62 ms
 3   2  26.1.1.1  4.41 ms
 3   3  26.1.1.1  4.64 ms
 4   1  26.1.1.2  5.74 ms
 4   2  26.1.1.2  6.22 ms
 4   3  26.1.1.2  5.77 ms
```

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:

- **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.

All TWAMP Light reserved UDP ports default to the OAM-PM application. To allocate a UDP port to an application, no other application can have the UDP source port configured under any test or template, regardless of administrative state.

> **Note:** 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 dynamic allocaiton of source UDP ports unless a specific technical reason is present.

4.1 OAM Performance Monitoring

Use the following command to configure the test packet type for the Session-Sender.

```
configure oam-pm session ip twamp-light session-sender-type
```

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 consists of 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

The following figure 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 25: OAM-PM architecture hierarchy

al_0386
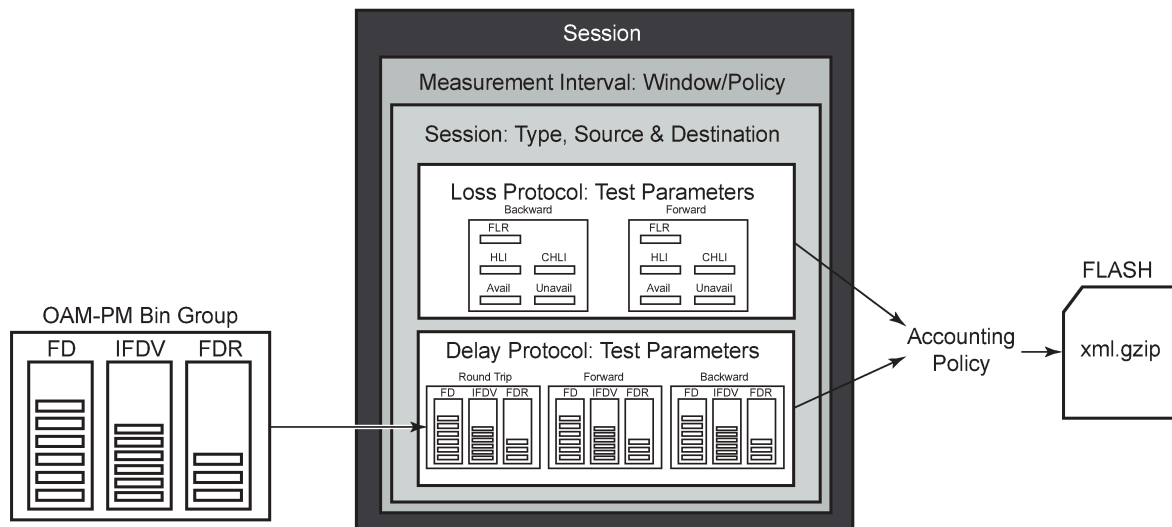
OAM-PM configurations are not dynamic environments. All aspects of the architecture must be carefully considered before configuring the various architectural components, making external references to other related components, or activating the OAM-PM architecture. No modifications are allowed to any 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 is 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.

4.1.1 Session

The session is the overall collection of test information fields. 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 are 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 test types and the technology-specific configuration under the session are based on the test family. Each of the test types requires a test ID as part of the configuration. When the test is configured using the **auto** keyword, the value is allocated at test create time. The test ID is released when the test is deleted. Any action that causes the test to be deleted and recreated releases the original test identifier and allocate a new one. These test identifiers are not persistent and not maintained across CPM switchovers. New test identifiers will be allocated in the case of CPM switchover.

- **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.

4.1.2 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.

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.

4.1.3 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 1 minute (**1-min**), 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 6: Measurement interval start times lists examples of the start times of each measurement interval.

Table 6: Measurement interval start times

Offset	1-min	15-min	1-hour	1-day
0 (default)	0, 1, 2, 3, ...	00, 15, 30, 45	00 (top of the hour)	midnight

Offset	1-min	15-min	1-hour	1-day
10 minutes	rejected	10, 25, 40, 55	10 min after the hour	10 min after midnight
30 minutes	rejected	rejected	30 min after the hour	30 min after midnight
60 minutes	rejected	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 7: OAM-PM XML keywords and MIB reference describes the keywords and MIB references contained in the OAM-PM XML file.

Table 7: OAM-PM XML keywords and MIB reference

XML file keyword	Description	TIMETRA-OAM-PM-MIB object
oampm	—	None - header only
Keywords shared by all OAM-PM protocols		
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
TLD	**TWAMP Light Delay Record**	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
TLL	**TWAMP Light Loss Record**	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	tmnxOamPmStsLossTwlUnavlIndFwd
uvb	Unavailable count in the backward direction	tmnxOamPmStsLossTwlUnavlIndBwd
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	tmnxOamPmStsLossTwlUndtUnavlFwd
uub	Undetermined unavailable count in the backward direction	tmnxOamPmStsLossTwlUndtUnavlBwd
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

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.

4.1.4 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:

- **Frame Delay**

  Frame Delay (FD) is the amount of time required to send and receive the packet.

- **InterFrame Delay Variation**

  InterFrame Delay Variation (IFDV) is the difference in the delay metrics between two adjacent packets.

- **Frame Delay Range**

  Frame Delay Range (FDR) is the difference between the minimum frame delay and the individual packet.

- **Mean Frame Delay**

  Mean Frame Delay (MFD) is 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 are 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.

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.

The following figure 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 26: Evaluating and computing loss and availability
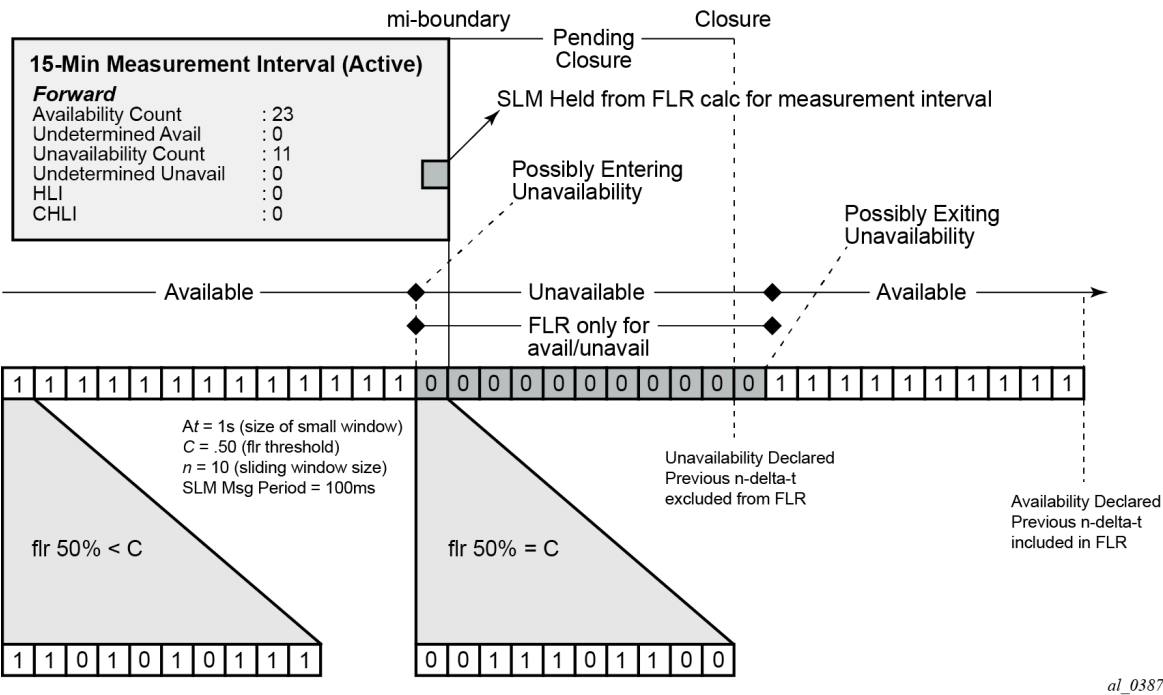
4.1.5 Bin groups

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.

Example: Bin group 1 configuration parameters

```
-------------------------------------------------------------------------------
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          -           -
-------------------------------------------------------------------------------
```

4.1.6 Relating the components

The following figure shows the architecture of all of the OAM-PM concepts previously discussed. It shows a more detailed hierarchy than previously shown in the introduction. This shows the relationship between the tests, the measurement intervals, and the storage of the results.

Figure 27: Relating OAM-PM components
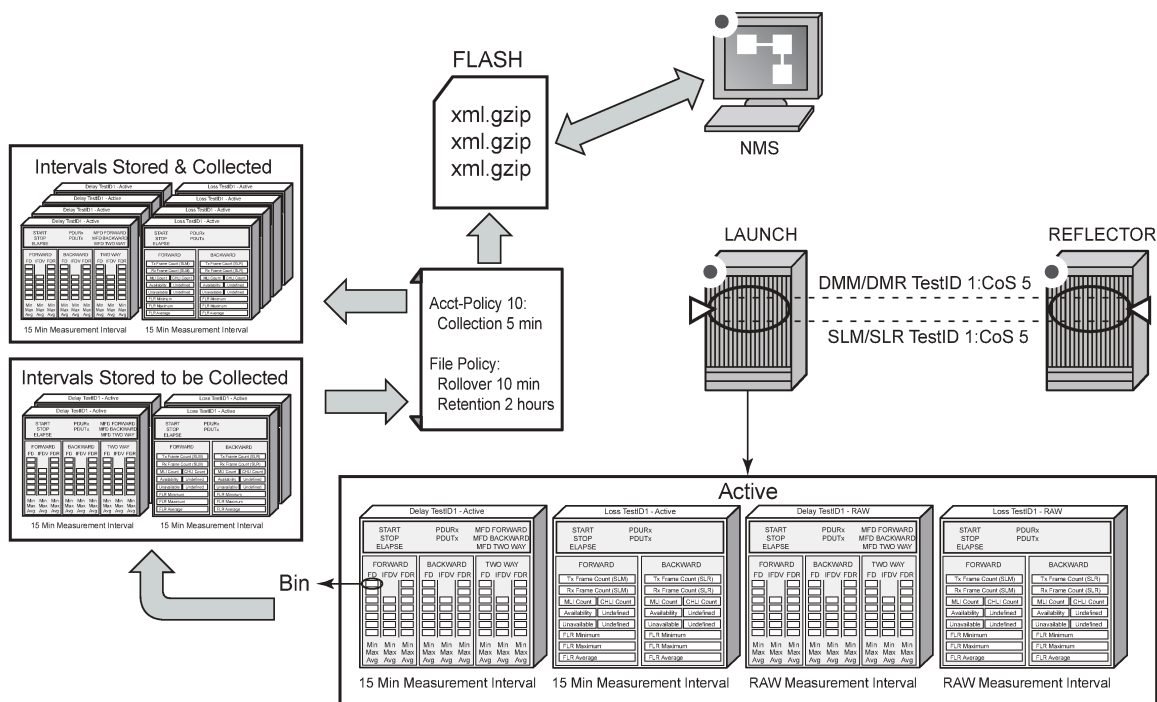
4.1.7 Monitoring

The following configuration examples demonstrate the show and monitoring commands available to check OAM-PM.

4.1.7.1 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 15-min Bins"
            record complete-pm
            collection-interval 5
            to file 1
            no shutdown
        exit
        log-id 1
        exit
----------------------------------------------
```

4.1.7.2 OAM-PM configuration

```
A:node-2>config>oam-pm# info detail
----------------------------------------------
        bin-group 1 fd-bin-count 3 fdr-bin-count 2 ifdv-bin-count 2 create
            description "OAM PM default bin group (not modifiable)"
            bin-type fd
                bin 1
                    lower-bound 5000
                exit
                bin 2
                    lower-bound 10000
                exit
            exit
            bin-type fdr
                bin 1
                    lower-bound 5000
                exit
            exit
            bin-type ifdv
                bin 1
                    lower-bound 5000
                exit
            exit
            no shutdown
        exit
        bin-group 2 fd-bin-count 10 fdr-bin-count 10 ifdv-bin-count 10 create
            description "max bins"
            bin-type fd
                bin 1
                    lower-bound 5000
                exit
                bin 2
                    lower-bound 10000
                exit
                bin 3
                    lower-bound 15000
                exit
                bin 4
```

```
                    lower-bound 20000
                exit
                bin 5
                    lower-bound 25000
                exit
                bin 6
                    lower-bound 30000
                exit
                bin 7
                    lower-bound 35000
                exit
                bin 8
                    lower-bound 40000
                exit
                bin 9
                    lower-bound 45000
                exit
                no delay-event forward
                no delay-event-exclusion forward
                no exclude-from-avg forward
                no delay-event backward
                no delay-event-exclusion backward
                no exclude-from-avg backward
                no delay-event round-trip
                no delay-event-exclusion round-trip
                no exclude-from-avg round-trip
            exit
            bin-type fdr
                bin 1
                    lower-bound 5000
                exit
                bin 2
                    lower-bound 10000
                exit
                bin 3
                    lower-bound 15000
                exit
                bin 4
                    lower-bound 20000
                exit
                bin 5
                    lower-bound 25000
                exit
                bin 6
                    lower-bound 30000
                exit
                bin 7
                    lower-bound 35000
                exit
                bin 8
                    lower-bound 40000
                exit
                bin 9
                    lower-bound 45000
                exit
                no delay-event forward
                no delay-event-exclusion forward
                no exclude-from-avg forward
                no delay-event backward
                no delay-event-exclusion backward
                no exclude-from-avg backward
                no delay-event round-trip
                no delay-event-exclusion round-trip
                no exclude-from-avg round-trip
```

```
                exit
            bin-type ifdv
                bin 1
                    lower-bound 5000
                exit
                bin 2
                    lower-bound 10000
                exit
                bin 3
                    lower-bound 15000
                exit
                bin 4
                    lower-bound 20000
                exit
                bin 5
                    lower-bound 25000
                exit
                bin 6
                    lower-bound 30000
                exit
                bin 7
                    lower-bound 35000
                exit
                bin 8
                    lower-bound 40000
                exit
                bin 9
                    lower-bound 45000
                exit
                no delay-event forward
                no delay-event-exclusion forward
                no exclude-from-avg forward
                no delay-event backward
                no delay-event-exclusion backward
                no exclude-from-avg backward
                no delay-event round-trip
                no delay-event-exclusion round-trip
                no exclude-from-avg round-trip
            exit
            no shutdown
        exit
        session "sess1" test-family ip session-type proactive create
            bin-group 2
            no description
            meas-interval 1-min create
                accounting-policy 11
                boundary-type clock-aligned
                clock-offset 0
                event-mon
                    shutdown
                    no delay-events
                    no loss-events
                exit
                intervals-stored 32
            exit
            meas-interval 5-mins create
                accounting-policy 22
                boundary-type clock-aligned
                clock-offset 0
                event-mon
                    shutdown
                    no delay-events
                    no loss-events
                exit
```

```
                intervals-stored 32
            exit
            meas-interval 15-mins create
                accounting-policy 2
                boundary-type clock-aligned
                clock-offset 0
                event-mon
                    shutdown
                    no delay-events
                    no loss-events
                exit
                intervals-stored 32
            exit
            meas-interval 1-hour create
                accounting-policy 3
                boundary-type clock-aligned
                clock-offset 0
                event-mon
                    shutdown
                    no delay-events
                    no loss-events
                exit
                intervals-stored 24
            exit
            meas-interval 1-day create
                accounting-policy 4
                boundary-type clock-aligned
                clock-offset 0
                event-mon
                    shutdown
                    no delay-events
                    no loss-events
                exit
                intervals-stored 1
            exit
            ip
                no allow-egress-remark-dscp
                dest-udp-port 64364
                destination 10.20.1.2
                no do-not-fragment
                dscp resolve
                fc "be"
                no forwarding
                pattern 0
                profile out
                router "Base"
                source 10.10.3.3
                no source-udp-port
                ttl 255
                tunnel
            exit
            twamp-light test-id 0 create
                session-sender-type twamp-light
                no allow-ipv6-udp-checksum-zero
                no delay-template
                interval 100
                loss
                    flr-threshold 50
                    no hli-force-count
                    timing frames-per-delta-t 1 consec-delta-t 10 chli-threshold 5
                exit
                loss-events
                    no avg-flr-event forward
                    no avg-flr-event backward
```

```
                            no chli-event forward
                            no hli-event forward
                            no unavailability-event forward
                            no undet-availability-event forward
                            no undet-unavailability-event forward
                            no chli-event backward
                            no hli-event backward
                            no unavailability-event backward
                            no undet-availability-event backward
                            no undet-unavailability-event backward
                            no chli-event aggregate
                            no hli-event aggregate
                            no unavailability-event aggregate
                            no undet-availability-event aggregate
                            no undet-unavailability-event aggregate
                        exit
                        pad-size 0
                        no pad-tlv-size
                        record-stats delay-and-loss
                        no test-duration
                        timestamp-format ntp
                        no shutdown
                    exit
                exit
            exit
            session "sess2" test-family ip session-type on-demand create
                bin-group 2
                no description
                meas-interval 1-min create
                    accounting-policy 11
                    boundary-type clock-aligned
                    clock-offset 0
                    event-mon
                        shutdown
                        no delay-events
                        no loss-events
                    exit
                    intervals-stored 32
                exit
                meas-interval 5-mins create
                    accounting-policy 22
                    boundary-type clock-aligned
                    clock-offset 0
                    event-mon
                        shutdown
                        no delay-events
                        no loss-events
                    exit
                    intervals-stored 32
                exit
                meas-interval 15-mins create
                    accounting-policy 2
                    boundary-type clock-aligned
                    clock-offset 0
                    event-mon
                        shutdown
                        no delay-events
                        no loss-events
                    exit
                    intervals-stored 32
                exit
                meas-interval 1-hour create
                    accounting-policy 3
                    boundary-type clock-aligned
```

```
                    clock-offset 0
                    event-mon
                        shutdown
                        no delay-events
                        no loss-events
                    exit
                    intervals-stored 24
                exit
                meas-interval 1-day create
                    accounting-policy 4
                    boundary-type clock-aligned
                    clock-offset 0
                    event-mon
                        shutdown
                        no delay-events
                        no loss-events
                    exit
                    intervals-stored 1
                exit
                ip
                    no allow-egress-remark-dscp
                    dest-udp-port 64364
                    destination 10.20.1.2
                    no do-not-fragment
                    dscp resolve
                    fc "be"
                    no forwarding
                    pattern 0
                    profile out
                    router "Base"
                    source 10.10.3.3
                    no source-udp-port
                    ttl 255
                    tunnel
                    exit
                    twamp-light test-id 1 create
                        session-sender-type twamp-light
                        no allow-ipv6-udp-checksum-zero
                        no delay-template
                        interval 100
                        loss
                            flr-threshold 50
                            no hli-force-count
                            timing frames-per-delta-t 1 consec-delta-t 10 chli-threshold 5
                        exit
                        loss-events
                            no avg-flr-event forward
                            no avg-flr-event backward
                            no chli-event forward
                            no hli-event forward
                            no unavailability-event forward
                            no undet-availability-event forward
                            no undet-unavailability-event forward
                            no chli-event backward
                            no hli-event backward
                            no unavailability-event backward
                            no undet-availability-event backward
                            no undet-unavailability-event backward
                            no chli-event aggregate
                            no hli-event aggregate
                            no unavailability-event aggregate
                            no undet-availability-event aggregate
                            no undet-unavailability-event aggregate
                        exit
```

```
                    pad-size 0
                    no pad-tlv-size
                    record-stats delay-and-loss
                    no test-duration
                    timestamp-format ntp
                    no shutdown
                exit
            exit
        exit
        streaming
        exit
```

4.1.7.3 Show and monitor commands

```
show oam-pm bin-group


-------------------------------------------------------------------------------
Configured Lower Bounds for Delay 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      max bins                    Up    0         0          0          0
                                          1      5000       5000       5000
                                          2     10000      10000      10000
                                          3     15000      15000      15000
                                          4     20000      20000      20000
                                          5     25000      25000      25000
                                          6     30000      30000      30000
                                          7     35000      35000      35000
                                          8     40000      40000      40000
                                          9     45000      45000      45000
-------------------------------------------------------------------------------
* indicates that the corresponding row element may have been truncated.
*A:Dut-C>config>oam-pm#

*A:Dut-C>config>oam-pm# show oam-pm bin-group-using


===============================================================================
OAM Performance Monitoring Bin Group Configuration for Sessions
===============================================================================
Bin Group       Admin    Session                          Session State
-------------------------------------------------------------------------------
2               Up       sess1                                 Act
                         sess2                                 Act
-------------------------------------------------------------------------------
===============================================================================
*A:Dut-C>config>oam-pm#

*A:Dut-C>config>oam-pm# show oam-pm sessions test-family ip


===============================================================================
OAM Performance Monitoring Session Summary for the IP Test Family
===============================================================================
Session                     State   Bin Group   Sess Type   Test Types
-------------------------------------------------------------------------------
sess1                       Act        2       proactive       TWL
sess2                       Act        2       on-demand       TWL
```

```
===============================================================================
*A:Dut-C>config>oam-pm#
*A:Dut-C>config>oam-pm# show oam-pm session "sess1" all

-------------------------------------------------------------------------------
Basic Session Configuration
-------------------------------------------------------------------------------
Session Name      : sess1
Description       : (Not Specified)
Test Family       : ip                      Session Type       : proactive
Bin Group         : 2
-------------------------------------------------------------------------------


-------------------------------------------------------------------------------
IP Configuration
-------------------------------------------------------------------------------
Source IP Address : 10.10.3.3
Dest IP Address   : 10.20.1.2
Confg Src UDP Port: (Not Specified)     In-Use Src UDP Port: 49154
Dest UDP Port     : 64364               Time To Live       : 255
Forwarding Class  : be                  Profile            : out
DSCP              : resolve             Allow Remark DSCP  : no
Router            : Base                Bypass Routing     : no
Egress Interface  : (Not Specified)
Next Hop Address  : (Not Specified)
Do Not Fragment   : no                  Pattern            : 0
Router Instnce:(Not Specified)
Tunnel Type       : none
-------------------------------------------------------------------------------


-------------------------------------------------------------------------------
TWAMP-Light Test Configuration and Status     Session Sender Type : TWAMP-Light
-------------------------------------------------------------------------------
Test ID           : 0                   Admin State        : Up
Oper State        : Up                  Pad Size           : 0 octets
Pad TLV Size      : Not Applicable      Timestamp Format   : NTP
On-Demand Duration: Not Applicable      On-Demand Remaining: Not Applicable
Interval          : 100 ms              Record Stats       : delay-and-loss
CHLI Threshold    : 5 HLIs              Frames Per Delta-T : 1 frames
Consec Delta-Ts   : 10                  FLR Threshold      : 50%
HLI Force Count   : no                  IPv6 UDP Checksum 0: Disallow
Detectable Tx Err : source IP address is not local
Session Sender ID : Not Applicable
STAMP U Flags Rx  : Not Applicable      STAMP M Flags Rx   : Not Applicable
Str Delay Tmpl:(Not Specified)
-------------------------------------------------------------------------------


-------------------------------------------------------------------------------
1-min Measurement Interval Configuration
-------------------------------------------------------------------------------
Duration          : 1-min               Intervals Stored   : 32
Boundary Type     : clock-aligned       Clock Offset       : 0 seconds
Accounting Policy : 11                  Event Monitoring   : disabled
Delay Event Mon   : disabled            Loss Event Mon     : disabled
-------------------------------------------------------------------------------


-------------------------------------------------------------------------------
5-mins Measurement Interval Configuration
-------------------------------------------------------------------------------
Duration          : 5-mins              Intervals Stored   : 32
Boundary Type     : clock-aligned       Clock Offset       : 0 seconds
Accounting Policy : 22                  Event Monitoring   : disabled
Delay Event Mon   : disabled            Loss Event Mon     : disabled
-------------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------
15-mins Measurement Interval Configuration
-------------------------------------------------------------------------
Duration          : 15-mins        Intervals Stored  : 32
Boundary Type     : clock-aligned  Clock Offset      : 0 seconds
Accounting Policy : 2              Event Monitoring  : disabled
Delay Event Mon   : disabled       Loss Event Mon    : disabled
-------------------------------------------------------------------------


-------------------------------------------------------------------------
1-hour Measurement Interval Configuration
-------------------------------------------------------------------------
Duration          : 1-hour         Intervals Stored  : 24
Boundary Type     : clock-aligned  Clock Offset      : 0 seconds
Accounting Policy : 3              Event Monitoring  : disabled
Delay Event Mon   : disabled       Loss Event Mon    : disabled
-------------------------------------------------------------------------


-------------------------------------------------------------------------
1-day Measurement Interval Configuration
-------------------------------------------------------------------------
Duration          : 1-day          Intervals Stored  : 1
Boundary Type     : clock-aligned  Clock Offset      : 0 seconds
Accounting Policy : 4              Event Monitoring  : disabled
Delay Event Mon   : disabled       Loss Event Mon    : disabled
-------------------------------------------------------------------------


-------------------------------------------------------------------------
Configured Lower Bounds for Delay Tests, in microseconds
-------------------------------------------------------------------------
Group Description               Admin Bin    FD(us)    FDR(us)   IFDV(us)
-------------------------------------------------------------------------
2     max bins                 Up   0           0          0          0
                                    1        5000       5000       5000
                                    2       10000      10000      10000
                                    3       15000      15000      15000
                                    4       20000      20000      20000
                                    5       25000      25000      25000
                                    6       30000      30000      30000
                                    7       35000      35000      35000
                                    8       40000      40000      40000
                                    9       45000      45000      45000
-------------------------------------------------------------------------


----------------------------------------
Bins Excluded from Average
----------------------------------------
Bin Type     Direction          Bins
----------------------------------------
----------------------------------------


----------------------------------------------------------------
Bins Excluded from Delay Event Count
----------------------------------------------------------------
Bin Type     Direction    Lowest Excluded Bin    Lower Bound (us)
----------------------------------------------------------------
----------------------------------------------------------------


-------------------------------------------------------------------------
Delay Events for the TWAMP-Light Test
-------------------------------------------------------------------------
Bin Type   Direction   LowerBound(us)   Raise   Clear       Last TCA (UTC)
-------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
Loss Events for the TWAMP-Light Test
-------------------------------------------------------------------------------
Event Type              Direction    Raise     Clear        Last TCA (UTC)
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
*A:Dut-C>config>oam-pm#
*A:Dut-C>config>oam-pm# show oam-pm statistics session "sess1" twamp-light meas-interval raw


-------------------------------------------------------------------------------
Start (UTC)       : 2025/02/20 21:49:32       Status       : in-progress
Elapsed (seconds) : 151                       Suspect      : yes
Frames Sent       : 1                         Frames Received : 0
-------------------------------------------------------------------------------
===============================================================================
TWAMP-LIGHT DELAY STATISTICS


-------------------------------------------------------------------------------
Bin Type    Direction    Minimum (us)  Maximum (us)  Average (us)   EfA
-------------------------------------------------------------------------------
FD          Forward              0             0             0      no
FD          Backward             0             0             0      no
FD          Round Trip           0             0             0      no
FDR         Forward              0             0             0      no
FDR         Backward             0             0             0      no
FDR         Round Trip           0             0             0      no
IFDV        Forward              0             0             0      no
IFDV        Backward             0             0             0      no
IFDV        Round Trip           0             0             0      no
-------------------------------------------------------------------------------
EfA = yes: one or more bins configured to be Excluded from the Average calc.


-------------------------------------------------------------
Frame Delay (FD) Bin Counts
-------------------------------------------------------------
Bin     Lower Bound     Forward      Backward     Round Trip
-------------------------------------------------------------
0              0 us        0             0             0
1           5000 us        0             0             0
2          10000 us        0             0             0
3          15000 us        0             0             0
4          20000 us        0             0             0
5          25000 us        0             0             0
6          30000 us        0             0             0
7          35000 us        0             0             0
8          40000 us        0             0             0
9          45000 us        0             0             0
-------------------------------------------------------------


-------------------------------------------------------------
Frame Delay Range (FDR) Bin Counts
-------------------------------------------------------------
Bin     Lower Bound     Forward      Backward     Round Trip
-------------------------------------------------------------
0              0 us        0             0             0
1           5000 us        0             0             0
2          10000 us        0             0             0
3          15000 us        0             0             0
4          20000 us        0             0             0
5          25000 us        0             0             0
6          30000 us        0             0             0
```

```
7              35000 us              0              0              0
8              40000 us              0              0              0
9              45000 us              0              0              0
-----------------------------------------------------------------


-----------------------------------------------------------------
Inter-Frame Delay Variation (IFDV) Bin Counts
-----------------------------------------------------------------
Bin      Lower Bound        Forward        Backward      Round Trip
-----------------------------------------------------------------
0                0 us              0              0              0
1             5000 us              0              0              0
2            10000 us              0              0              0
3            15000 us              0              0              0
4            20000 us              0              0              0
5            25000 us              0              0              0
6            30000 us              0              0              0
7            35000 us              0              0              0
8            40000 us              0              0              0
9            45000 us              0              0              0
-----------------------------------------------------------------
=================================================================
=================================================================
TWAMP-LIGHT LOSS STATISTICS

-------------------------------------------------------
                  Frames Sent        Frames Received
-------------------------------------------------------
Forward                   0                      0
Backward                  0                      0
-------------------------------------------------------


-----------------------------------------------
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        1459        1459           0           0        0         0
Backward       1459        1459           0           0        0         0
--------------------------------------------------------------------------
=================================================================
*A:Dut-C>config>oam-pm#
```

The **monitor** command can be used to automatically update the statistics for the raw measurement
interval.

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 Bidirectional Forwarding Detection (BFD)

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)*

5.2 Border Gateway Protocol (BGP)

draft-hares-idr-update-attrib-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-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-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-link-bandwidth-03, *BGP Link Bandwidth Extended Community*

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 6368, *Internal BGP as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*

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 7752, *North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP*

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 9494, *Long-Lived Graceful Restart for BGP*

5.3 Bridging and management

IEEE 802.1AB, *Station and Media Access Control Connectivity Discovery*

IEEE 802.1ad, *Provider Bridges*

IEEE 802.1AX, *Link Aggregation*

IEEE 802.1Q, *Virtual LANs*

5.4 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*

5.5 Ethernet VPN (EVPN)

draft-ietf-bess-bgp-srv6-args-00, *SRv6 Argument Signaling for BGP Services*

draft-ietf-bess-evpn-ipvpn-interworking-06, *EVPN Interworking with IPVPN*

draft-sr-bess-evpn-vpws-gateway-03, *Ethernet VPN Virtual Private Wire Services Gateway Solution*

RFC 7432, *BGP MPLS-Based Ethernet VPN*

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 9047, *Propagation of ARP/ND Flags in an Ethernet Virtual Private Network (EVPN)*

RFC 9135, *Integrated Routing and Bridging in Ethernet VPN (EVPN)*

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)*

5.6 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*

gnmi_ext.proto version 0.1.0, *gNMI Commit Confirmed Extension*

system.proto version 1.0.0, *gRPC Network Operations Interface (gNOI) System Service*

PROTOCOL-HTTP2, *gRPC over HTTP2*

5.7 Intermediate System to Intermediate System (IS-IS)

draft-ietf-isis-mi-02, *IS-IS Multi-Instance*

draft-ietf-lsr-igp-ureach-prefix-announce-01, *IGP Unreachable Prefix Announcement* – without U-Flag and UP-Flag

draft-kaplan-isis-ext-eth-02, *Extended Ethernet Frame Size Support*

ISO/IEC 10589:2002 Second Edition, *Intermediate system to Intermediate system intra-domain routeing 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 6213, *IS-IS BFD-Enabled TLV*

RFC 6232, *Purge Originator Identification TLV for IS-IS*

RFC 6233, *IS-IS Registry Extension for Purges*

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*

5.8 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 2784, *Generic Routing Encapsulation (GRE)*

RFC 3164, *The BSD syslog Protocol*

RFC 4632, *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*

RFC 5082, *The Generalized TTL Security Mechanism (GTSM)*

5.9 Internet Protocol (IP) multicast

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 3376, *Internet Group Management Protocol, Version 3*

RFC 3446, *Anycast Rendevous 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 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 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 7761, *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*

RFC 8487, *Mtrace Version 2: Traceroute Facility for IP Multicast*

5.10 Internet Protocol (IP) version 4

RFC 791, *Internet Protocol*

RFC 792, *Internet Control Message Protocol*

RFC 826, *An Ethernet Address Resolution Protocol*

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 1812, *Requirements for IPv4 Routers*

RFC 1918, *Address Allocation for Private Internets*

RFC 2131, *Dynamic Host Configuration Protocol*; Relay only

RFC 2132, *DHCP Options and BOOTP Vendor Extensions* – DHCP

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

5.11 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 3736, *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6*

RFC 3971, *SEcure Neighbor Discovery (SEND)*

RFC 4007, *IPv6 Scoped Address Architecture*

RFC 4191, *Default Router Preferences and More-Specific Routes* – Default Router Preference

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 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 6164, *Using 127-Bit IPv6 Prefixes on Inter-Router Links*

RFC 8021, *Generation of IPv6 Atomic Fragments Considered Harmful*

RFC 8200, *Internet Protocol, Version 6 (IPv6) Specification*

5.12 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)*

RFC 8784, *Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security*

5.13 Label Distribution Protocol (LDP)

draft-pdutta-mpls-ldp-adj-capability-00, *LDP Adjacency Capabilities*

draft-pdutta-mpls-ldp-v2-00, *LDP Version 2*

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*

5.14 Multiprotocol Label Switching (MPLS)

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 5332, *MPLS Multicast Encapsulations*

RFC 5884, *Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)*

RFC 6424, *Mechanism for Performing Label Switched Path Ping (LSP Ping) over MPLS Tunnels*

RFC 7308, *Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)*

RFC 7746, *Label Switched Path (LSP) Self-Ping*

5.15 Network Address Translation (NAT)

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*

5.16 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 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 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 5642, *Dynamic Hostname Exchange Mechanism for OSPF*

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 8920, *OSPF Application-Specific Link Attributes*

5.17 Path Computation Element Protocol (PCEP)

draft-alvarez-pce-path-profiles-04, *PCE Path Profiles*

draft-ietf-pce-binding-label-sid-15, *Carrying Binding Label/Segment Identifier (SID) in PCE-based Networks.* – MPLS binding SIDs

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*

5.18 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*

5.19 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)*

5.20 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 3162, *RADIUS and IPv6*

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*

5.21 Resource Reservation Protocol - Traffic Engineering (RSVP-TE)

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 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 5817, *Graceful Shutdown in MPLS and Generalized MPLS Traffic Engineering Networks*

5.22 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*

5.23 Segment Routing (SR)

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 8426, *Recommendations for RSVP-TE and Segment Routing (SR) Label Switched Path (LSP) Coexistence*

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 8665, *OSPF Extensions for Segment Routing*

RFC 8667, *IS-IS Extensions for Segment Routing*

RFC 8669, *Segment Routing Prefix Segment Identifier Extensions for BGP*

RFC 9256, *Segment Routing Policy Architecture*

RFC 9350, *IGP Flexible Algorithm*

5.24 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-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-vrrp-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*

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 2206, *RSVP Management Information Base using SMIv2*

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 3273, *Remote Network Monitoring Management Information Base for High Capacity Networks*

RFC 3410, *Introduction and Applicability Statements for Internet Standard Management Framework*

RFC 3430, *Simple Network Management Protocol (SNMP) over Transmission Control Protocol (TCP) Transport Mapping*

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 3584, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*

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 3826, *The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model*

RFC 3877, *Alarm Management Information Base (MIB)*

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 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 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*

5.25 Timing

RFC 3339, *Date and Time on the Internet: Timestamps*

RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*

RFC 8573, *Message Authentication Code for the Network Time Protocol*

5.26 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

RFC 9503, *Simple Two-Way Active Measurement Protocol (STAMP) Extensions for Segment Routing Networks* – excluding Sections 3, 4.1.2 and 4.1.3

5.27 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 7117, *Multicast in Virtual Private LAN Service (VPLS)*

5.28 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*

Customer document and product support

Customer documentation
Customer documentation welcome page

Technical support
Product support portal

Documentation feedback
Customer documentation feedback