Implementing MPLS OAM

IP-Less MPLS-TP Ping and MPLS-TP Traceroute

In Label Switched Path (LSP) ping or traceroute with IP encapsulation over ACH, IP encapsulated ping or traceroute packets are sent over the MPLS LSP using the control channel (ACH). The application-level control channel in this case is the reverse path of the LSP using ACH. The on-demand ping or traceroute echo response message is sent on the reverse path of the LSP. The response uses ACH and is IP encapsulated. The destination address in the IP header is set to that of the sender of the echo request message, and the source address in the IP header is set to a valid address of the replying node.

- the reply mode is 4

- the node does not have a return MPLS LSP path to the echo request source.

MPLS LSP Ping

The MPLS LSP Ping feature is used to check the connectivity between Ingress LSR and egress LSRs along an LSP. MPLS LSP ping uses MPLS echo request and reply messages, similar to Internet Control Message Protocol (ICMP) echo request and reply messages, to validate an LSP. While ICMP echo request and reply messages validate IP networks, MPLS echo and reply messages validate MPLS networks. The MPLS echo request packet is sent to a target router through the use of the appropriate label stack associated with the LSP to be validated. Use of the label stack causes the packet to be forwarded over the LSP itself. The destination IP address of the MPLS echo request packet is different from the address used to select the label stack. The destination IP address is defined as a 127.x.y.z/8 address and it prevents the IP packet from being IP switched to its destination, if the LSP is broken.

An MPLS echo reply is sent in response to an MPLS echo request. The reply is sent as an IP packet and it is forwarded using IP, MPLS, or a combination of both types of switching. The source address of the MPLS echo reply packet is an address obtained from the router generating the echo reply. The destination address

is the source address of the router that originated the MPLS echo request packet. The MPLS echo reply destination port is set to the echo request source port.

The following figure shows MPLS LSP ping echo request and echo reply paths.

Figure 1: MPLS LSP Ping Echo Request and Reply Paths
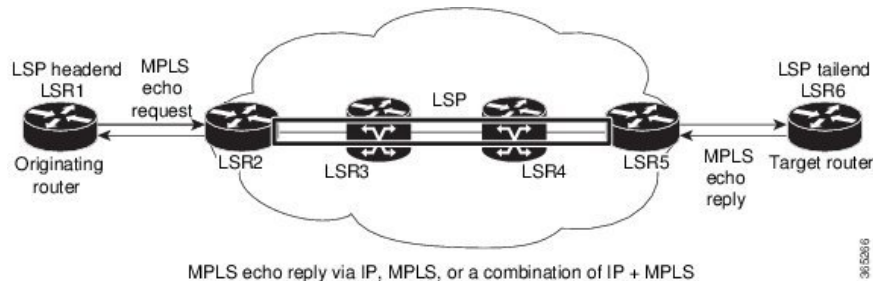
By default, the **ping mpls ipv4** command tries to determine the Forwarding Equivalence Class (FEC) being used automatically. However, this is only applicable at head-end and works only if the FEC at the destination is same as the source. If the source and destination FEC types are not the same, the **ping mpls ipv4** command may fail to identify the targeted FEC type. You can overcome this limitation by specifying the FEC type in MPLS LSP ping using the **fec-type** command option. If the user is not sure about the FEC type at the transit or the destination, or it may change through network, use of the **generic** FEC type command option is recommended. Generic FEC is not coupled to a particular control plane and allows path verification when the advertising protocol is unknown, or may change during the path of the echo request. If you are aware of the destination FEC type, specify the target FEC as BGP or LDP.

Configuration Examples

This example shows how to use MPLS LSP ping to test the connectivity of an IPv4 LDP LSP. The destination is specified as a Label Distribution Protocol (LDP) IPv4 address.

```
RP/0/RP0/CPU0:router# ping mpls ipv4 10.1.1.2/32 verbose

Sun Nov 15 11:27:43.070 UTC

Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
      timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

!      size 100, reply addr 10.1.0.2, return code 3
!      size 100, reply addr 10.1.0.2, return code 3
!      size 100, reply addr 10.1.0.2, return code 3
!      size 100, reply addr 10.1.0.2, return code 3
!      size 100, reply addr 10.1.0.2, return code 3

Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/4 ms
```

In this example, the destination is specified as a Label Distribution Protocol (LDP) IPv4 prefix and Forwarding Equivalence Class (FEC) type is specified as generic.

```
RP/0/RP0/CPU0:router# ping mpls ipv4 10.1.1.2/32 fec-type generic

Wed Nov 25 03:36:33.143 UTC
Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
      timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

In this example, the destination is specified as a Label Distribution Protocol (LDP) IPv4 prefix and the FEC type is specified as BGP.

```
RP/0/RP0/CPU0:router# ping mpls ipv4 10.1.1.2/32 fec-type bgp

Wed Nov 25 03:38:33.143 UTC
Sending 5, 100-byte MPLS Echos to 10.1.1.2/32,
      timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

MPLS LSP Traceroute

The MPLS LSP Traceroute feature is used to isolate the failure point of an LSP. It is used for hop-by-hop fault localization and path tracing. The MPLS LSP Traceroute feature relies on the expiration of the Time to Live (TTL) value of the packet that carries the echo request. When the MPLS echo request message hits a transit node, it checks the TTL value and if it is expired, the packet is passed to the control plane, else the message is forwarded. If the echo message is passed to the control plane, a reply message is generated based on the contents of the request message.

The following figure shows an MPLS LSP traceroute example with an LSP from LSR1 to LSR4.

Figure 2: MPLS LSP Traceroute
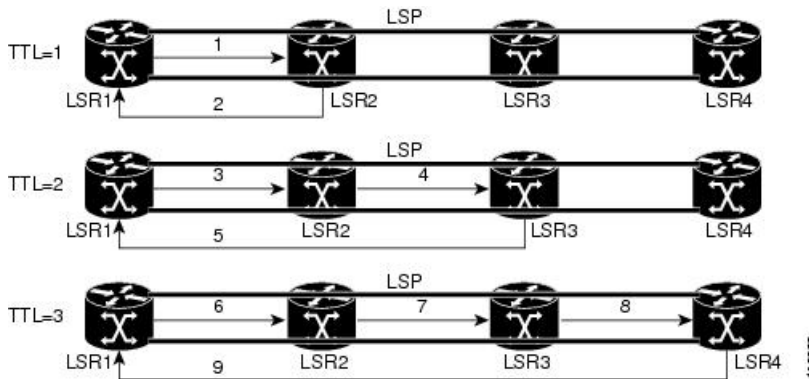
By default, the **traceroute mpls ipv4** command tries to determine the Forwarding Equivalence Class (FEC) being used automatically. However, this is only applicable at head-end and works only if the FEC at the destination is same as the source. If the source and destination FEC types are not the same, the **traceroute mpls ipv4** command may fail to identify the targeted FEC type. You can overcome this limitation by specifying the FEC type in MPLS LSP traceroute using the **fec-type** command option. If the user is not sure about the FEC type at the transit or the destination, or it may change through network, use of the **generic** FEC type command option is recommended. Generic FEC is not coupled to a particular control plane and allows path verification when the advertising protocol is unknown, or may change during the path of the echo request. If you are aware of the destination FEC type, specify the target FEC as BGP or LDP.

Configuration Examples

This example shows how to use the **traceroute** command to trace to a destination.

```
RP/0/RP0/CPU0:router# traceroute mpls ipv4 10.1.1.2/32 destination 127.0.0.3 127.0.0.6 2
Sat Jan 27 03:50:23.746 UTC

Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

Destination address 127.0.0.3
  0 10.2.1.2 MRU 1500 [Labels: 24000 Exp: 0]
L 1 10.2.1.1 MRU 1500 [Labels: implicit-null Exp: 0] 8 ms
! 2 10.1.0.2 3 ms

Destination address 127.0.0.5
  0 10.2.1.2 MRU 1500 [Labels: 24000 Exp: 0]
L 1 10.2.1.1 MRU 1500 [Labels: implicit-null Exp: 0] 5 ms
! 2 10.1.0.2 2 ms
```

This example shows how to use the **traceroute** command and how to specify the maximum number of hops for the traceroute to traverse by specifying the **ttl** value.

```
RP/0/RP0/CPU0:router# traceroute mpls ipv4 10.1.1.2/32 ttl 1
Sun Nov 15 12:20:14.145 UTC
Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.

  0 10.1.0.1 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 10.1.0.2 3 ms
```

This example shows how to use the **traceroute** command to trace to a destination and FEC type is specified as generic.

```
RP/0/RP0/CPU0:router# traceroute mpls ipv4 10.1.1.2/32 fec-type generic
Sun Nov 15 12:25:14.145 UTC
Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'L' - labeled output interface, 'B' - unlabeled output interface,
'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
'P' - no rx intf label prot, 'p' - premature termination of LSP,
'R' - transit router, 'I' - unknown upstream index,
'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
0 10.12.12.1 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 10.12.12.2 2 ms
```

This example shows how to use the **traceroute** command to trace to a destination and FEC type is specified as BGP.

```
RP/0/RP0/CPU0:router# traceroute mpls ipv4 10.1.1.2/32 fec-type bgp
Sun Nov 15 12:25:14.145 UTC
Tracing MPLS Label Switched Path to 10.1.1.2/32, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
'L' - labeled output interface, 'B' - unlabeled output interface,
'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
'M' - malformed request, 'm' - unsupported tlvs, 'N' - no rx label,
'P' - no rx intf label prot, 'p' - premature termination of LSP,
'R' - transit router, 'I' - unknown upstream index,
'X' - unknown return code, 'x' - return code 0
Type escape sequence to abort.
0 10.12.12.1 MRU 1500 [Labels: implicit-null Exp: 0]
! 1 10.12.12.2 2 ms
```

MPLS OAM Using Nil FEC

The Nil-FEC LSP ping and traceroute operations are extensions of regular MPLS ping and traceroute. MPLS ping and traceroute requires at least one forwarding equivalence class (FEC) in the target FEC stack. In Nil-FEC ping and traceroute operations, an explicit FEC is not associated with the label. Nil-FEC LSP ping and traceroute support MPLS static LSPs and also act as an additional diagnostic tool for all other LSP types.

Nil-FEC LSP ping and traceroute allow network operators to provide the ability to freely test any label stack by allowing them to specify the following:

- label stack

- outgoing interface

- nexthop address

The following table shows the syntax for the ping and traceroute commands.

Table 1: LSP Ping and Traceroute Nil FEC Commands

Command Syntax
ping mpls nil-fec labels {*label*[*,label*]} [**output** {**interface** *tx-interface*} [**nexthop** *nexthop-ip-addr*]]
traceroute mpls nil-fec labels {*label*[*,label*]} [**output** {**interface** *tx-interface*} [**nexthop** *nexthop-ip-addr*]]

Examples: LSP Ping Nil FEC and LSP Traceroute Nil FEC

The examples in this section use the following topology:

```
Node loopback IP address: 172.18.1.3   172.18.1.4   172.18.1.5   172.18.1.7
Node label:               16003        16004        16005        16007
Nodes:                    Arizona ---- Utah ------- Wyoming ---- Texas

Interface:           GigabitEthernet0/2/0/1   GigabitEthernet0/2/0/1
Interface IP address:      10.1.1.3                 10.1.1.4


RP/0/RP0/CPU0:router-arizona# show mpls forwarding

Tue May  2 13:44:31.999 EDT
Local  Outgoing    Prefix            Outgoing     Next Hop        Bytes
Label  Label       or ID             Interface                    Switched
------ ----------- ----------------- ------------ --------------- ------------
16004  Pop         No ID             Gi0/2/0/1    10.1.1.4        1392
       Pop         No ID             Gi0/2/0/2    10.1.2.2        0
16005  16005       No ID             Gi0/2/0/0    10.1.1.4        0
       16005       No ID             Gi0/2/0/1    10.1.2.2        0
16007  16007       No ID             Gi0/2/0/0    10.1.1.4        4752
       16007       No ID             Gi0/2/0/1    10.1.2.2        0
```

This example shows how to use Nil-FEC LSP ping to test a label stack.

```
RP/0/RP0/CPU0:router-arizona# ping mpls nil-fec labels 16005,16007 output interface
GigabitEthernet 0/2/0/1 nexthop 10.1.1.4 repeat 1
Sending 1, 72-byte MPLS Echos with Nil FEC labels 16005,16007,
    timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
```

```
   'd' - see DDMAP for return code,
   'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
!
Success rate is 100 percent (1/1), round-trip min/avg/max = 1/1/1 ms
 Total Time Elapsed 0 ms
```

This example shows how to use Nil-FEC LSP traceroute for a label stack.

```
RP/0/RP0/CPU0:router-arizona# traceroute mpls nil-fec labels 16005,16007 output interface
GigabitEthernet 0/2/0/1 nexthop 10.1.1.4
Tracing MPLS Label Switched Path with Nil FEC labels 16005,16007, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
   'L' - labeled output interface, 'B' - unlabeled output interface,
   'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
   'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
   'P' - no rx intf label prot, 'p' - premature termination of LSP,
   'R' - transit router, 'I' - unknown upstream index,
   'd' - see DDMAP for return code,
   'X' - unknown return code, 'x' - return code 0

Type escape sequence to abort.
  0 10.1.1.3 MRU 1500 [Labels: 16005/16007/explicit-null Exp: 0/0/0]
L 1 10.1.1.4 MRU 1500 [Labels: implicit-null/16007/explicit-null Exp: 0/0/0] 1 ms
L 2 10.1.1.5 MRU 1500 [Labels: implicit-null/explicit-null Exp: 0/0] 1 ms
! 3 10.1.1.7 1 ms
```

Self-Ping Probe for Reoptimized LSP

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
Self-Ping Probe for Reoptimized LSP	Release 7.5.3	You can now prevent traffic drops on a reoptimized label switch path (LSP) by timely confirmation that it's ready to handle the traffic. This confirmation is made possible by enabling the label edge router (LER) to send self-ping probes over the reoptimized LSP to the ingress LER. As soon the probe reaches the LER, there's confirmation that the RSVP programming is complete along the path. Post this confirmation, the LER switches traffic to the reoptimized LSP with no drop in traffic. This feature introduces the **self-ping** keyword in the **named-tunnels tunnel-te** command.

During the reoptimization of LSP, the label edge router (LER) has to wait until the reoptimized path is established before switching the traffic onto the reoptimized LSP. If the LER does not wait long and the RSVP programming for establishing MPLS forwarding along the path is not complete at a given hop along the path, then traffic is dropped at that hop. An LER is a router that operates at the edge of an MPLS network and acts as the entry and exit points for the network.

Your edge router's hardware has an internal *reopt-install timer* that grants the reoptimized path enough time to set up and then switches traffic to the reoptimized path. If the LER waits too long before switching the traffic, then the traffic is still sent over the old path which can cause congestion. For now, operators use a conservative time to wait for switching the traffic to avoid traffic loss over the reoptimized LSP. Although this *reopt-install timer* is widely used in RSVP-TE, it still causes the old path to be used while the reoptimized path is fully ready, and could potentially cause traffic loss if some node along the path takes a longer time to program MPLS forwarding.

Self-Ping Probe Workflow

- The LER does not start the *reopt-install timer*.

- The LER sends the self-ping probe over the reoptimized LSP. By default, the frequency of the self-ping probe is one probe per second. The frequency at which the self-ping probes are sent is not configurable.

- When the probe is received, the LER declares that the LSP is UP and then triggers the switchover of the tunnel traffic from the old LSP to the reoptimized LSP. The LER does not send any more probes over this LSP.

Key Concepts

Reoptimization occurs when a device with traffic engineering tunnels periodically examines tunnels with established LSPs to learn whether LSPs with better metrics are available and signal to that LSP. After the signaling is successful, the device replaces the older LSP with the reoptimized LSP.

In RSVP-TE networks, make-before-break or reoptimization is the mechanism of replacing an existing LSP without affecting the traffic that is carried on this LSP. The procedure in make-before-break is as follows:

- Create and signal a reoptimized LSP.

- Wait for this LSP to be ready to carry traffic or for the hardware programming to be complete at every hop in the path.

- Switch the traffic from the existing LSP to the newly created LSP.

- Wait for the hardware programming to be complete at the head, and then delete the existing or old LSP.

Configure Self-Ping Probe

Perform the following tasks to configure self-ping probe:

- Configure self-ping probe

- Configure maximum number of probes

Configuration Example

```
/* Configure Self-ping Probe */
Self-ping is supported for named-tunnels. The new keyword self-ping enables self-ping probe
 when tunnel-te ABC is being reoptimized.
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# named-tunnels tunnel-te ABC
Router(config-te-tun-name)# self-ping
Router(config-te-tun-name)# commit
```

```
/* Configure maximum number of probes */
You can configure the maximum number of probes before self-ping is considered unsuccessful.
Router# configure
Router(config)# mpls traffic-eng
Router(config-mpls-te)# named-tunnels tunnel-te ABC
Router(config-te-tun-name)# self-ping
Router(config-te-tun-name)# max-count 10
Router(config-te-tun-name)# commit
```

Running Configuration

This section shows the running configuration of self-ping probe.

```
/* Self-ping probe configuration */
mpls traffic-eng
  named-tunnels
    tunnel-te ABC
      self-ping
     !

/* Configure maximum number of probes */
mpls traffic-eng
  named-tunnels
    tunnel-te ABC
      self-ping
        max-count 10
     !
```

Verification

Verify the self-ping probe configuration.

```
Router# show mpls traffic-eng tunnels name ABC detail

Name: ABC  Destination: 192.168.0.4  Ifhandle:0x2d0
  Tunnel-ID: 32783
  Status:
    Admin:    up Oper:   up   Path:  valid   Signalling: connected

    G-PID: 0x0800 (derived from egress interface properties)
    Bandwidth Requested: 0 kbps  CT0
    Creation Time: Thu Apr  7 14:54:30 2022 (00:00:01 ago)
  Config Parameters:
    Bandwidth:         0 kbps (CT0) Priority:  7  7 Affinity: 0x0/0xffff
    Metric Type: TE (global)
    Path Selection:
      Tiebreaker: Min-fill (default)
    Hop-limit: disabled
    Cost-limit: disabled
    Delay-limit: disabled
    Delay-measurement: disabled
    Path-invalidation timeout: 10000 msec (default), Action: Tear (default)
    AutoRoute: disabled  LockDown: disabled   Policy class: not set
    Forward class: 0 (not enabled)
    Forwarding-Adjacency: disabled
    Autoroute Destinations: 0
    Loadshare:         0 equal loadshares
    Load-interval: 300 seconds
    Auto-bw: disabled
    Auto-Capacity: Disabled:
```

```
      Self-ping: Enabled
        Maximum-probes: 10
        Probes-period: 1 second(s)
      Fast Reroute: Disabled, Protection Desired: None
      Path Protection: Not Enabled
      BFD Fast Detection: Disabled
      Reoptimization after affinity failure: Enabled
      Soft Preemption: Disabled
    Self-ping:
      Status: Not executed
    SNMP Index: 0
    Binding SID: 0
    History:
      Tunnel has been up for: 00:00:00 (since Thu Jan 01 01:00:00 BST 1970)
      Current LSP:
        Uptime: 00:00:00 (since Thu Jan 01 01:00:00 BST 1970)
    Current LSP Info:
      Instance: 2, Signaling Area: OSPF 100 area 0
      Uptime: 00:00:00 (since Thu Apr 07 14:54:31 BST 2022)
      Outgoing Interface: GigabitEthernet0/2/0/0, Outgoing Label: 24000
      Next Hop: 10.10.10.2      Neighbor Next Hop: 0.0.0.0
      Router-IDs: local       192.168.0.1
                  downstream 192.168.0.2
      Soft Preemption: None
      SRLGs: not collected

        Record Route: Disabled
        Tspec: avg rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
        Session Attributes: Local Prot: Not Set, Node Prot: Not Set, BW Prot: Not Set
                            Soft Preemption Desired: Not Set
      Resv Info: None
        Record Route: Disabled
        Fspec: avg rate=0 kbits, burst=1000 bytes, peak rate=0 kbits
    Persistent Forwarding Statistics:
      Out Bytes: 0
      Out Packets: 0

Displayed 1 (of 1) heads, 0 (of 0) midpoints, 0 (of 0) tails
Displayed 1 up, 0 down, 0 recovering, 0 recovered heads
```

Verify the self-ping probe statistics.

```
Router# show mpls traffic-eng self-ping statistics
Wed Jun 15 14:24:29.316 BST
Self-Ping Statistics:
  Collected since: Tue Jun 14 09:35:52 2022 (1d04h ago)
  Operations:
    Started 2
    Running 0
    Successful 1
    Timed-out 1
    Terminated 0
  Probes sent 11
  Probes failed 0
  Received responses 1 (Average response time 00:00:00)
  Mismatched responses 0
```

During the self-ping procedure, the tunnel may go through various stages and the LER shows the status when you run the **show mpls traffic-eng tunnels detail** command.

- Though you have configured the self-pring feature, there is no request for reoptimization and the status is shown as *Not executed*.

```
Self-ping:
    Status: Not executed
```

- The reoptimization is ongoing and the self-ping is actively sending probes and waiting to receive the response and status is shown as *Underway*.

```
Self-ping:
    Status: Underway
    LSP-ID: 6
    Started: 00:00:00 (since Thu Apr 07 15:03:33 BST 2022)
```

- When the self-ping receives the response and the traffic is switched to the reoptimized LSP. The status is shown as *Succeeded*.

```
Self-ping:
    Status: Succeeded (in 0 seconds)
    LSP-ID: 4
    Probes sent: 1
    Started: 00:00:00 (since Thu Apr 07 15:02:23 BST 2022)
    Stopped: 00:00:00 (since Thu Apr 07 15:02:23 BST 2022)
    Response Received: 00:00:00 (since Thu Apr 07 15:02:23 BST 2022)
```

- When the LER sends the self-ping probe and did not receive any response. The status is shown as *Timed-out*.

  The LER falls back to the *reopt-install timer* and LSP is considered active, and traffic is switched to the reoptimized LSP after the *reopt-install timer* expires.

```
  Self-ping:
    Status: Timed-out (in 9 seconds)
    LSP-ID: 6
    Probes sent: 10
    Started: 00:00:43 (since Thu Apr 07 15:03:33 BST 2022)
    Stopped: 00:00:34 (since Thu Apr 07 15:03:42 BST 2022)
```

- The LER terminates self-ping execution when:

    - The reoptimized LSP has failed.

    - The *reopt-install timer* expires faster while the self-ping is still actively sending probes and receiving no response.

  The status is shown as *Terminated*.

```
Self-ping:
    Status: Terminated (in 20 seconds)
    LSP-ID: 8
    Started: 00:00:21 (since Thu Apr 07 15:05:44 BST 2022)
    Stopped: 00:00:01 (since Thu Apr 07 15:06:04 BST 2022)
```