# Configure Performance Measurement

Network performance metrics such as packet loss, delay, delay variation, and bandwidth utilization is a critical measure for traffic engineering (TE) in service provider networks. These network performance metrics provide network operators information about the performance characteristics of their networks for performance evaluation and helps to ensure compliance with service level agreements. The service-level agreements (SLAs) of service providers depend on the ability to measure and monitor these network performance metrics. Network operators can use performance measurement (PM) feature to monitor the network metrics for links as well as end-to-end TE label switched paths (LSPs).

The following table explains the functionalities supported by performance measurement feature for measuring delay for links or SR policies.

***Table 1: Performance Measurement Functionalities***

| Functionality | Details |
| --- | --- |
| Profiles | You can configure different profiles for different types of delay measurements. Delay profile type interfaces is used for link-delay measurement. Delay profile type sr-policy is used for SR policy delay measurements. Delay profile allows you to schedule probe and configure metric advertisement parameters for delay measurement. |
| Protocols | Protocols for delay measurement probes can be MPLS (using RFC6374 with MPLS encap) or TWAMP Light (using RFC 5357 with IP/UDP encap). |
| Probe and burst scheduling | Schedule probes and configure metric advertisement parameters for delay measurement. |
| Metric advertisements | Advertise measured metrics periodically using configured thresholds. Also supports accelerated advertisements using configured thresholds. |
| Measurement history and counters | Maintain packet delay and loss measurement history and also session counters and packet advertisement counters. |

# Measurement Modes

The following table compares the different hardware and timing requirements for the measurement modes supported in SR PM.

| Feature Name | Release | Description |
|---|---|---|
| SR Performance Measurement: Loopback Measurement Mode | Release 7.5.2 | Loopback measurement mode provides two-way and one-way measurements. PTP-capable hardware and hardware timestamping are required on the Sender but are not required on the Reflector.<br><br>Liveness monitoring uses "self-addressed" PM IP packets crafted by the sender (where the destination address is the sender's own IP address); this mode of operation is referred as "loopback mode”. |

***Table 2: Measurement Mode Requirements***

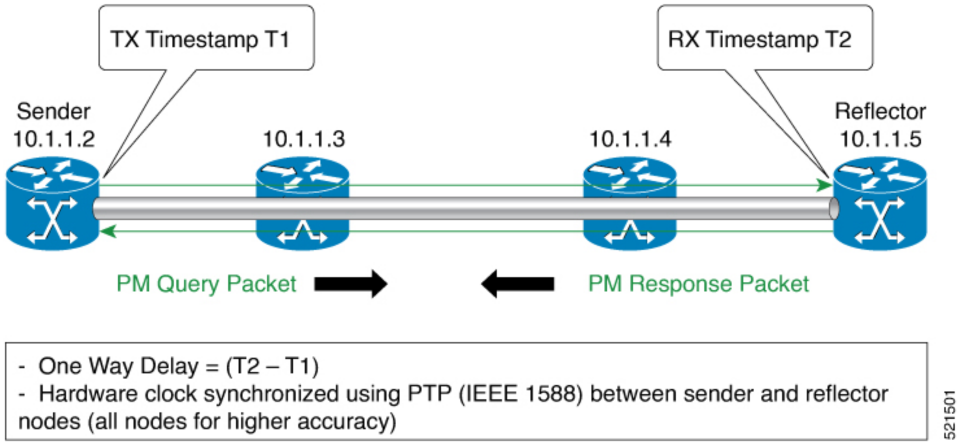
| Measurement Mode | Sender:<br><br>PTP-Capable HW and HW Timestamping | Reflector:<br><br>PTP-Capable HW and HW Timestamping | PTP Clock Synchronization between Sender and Reflector |
|---|---|---|---|
| One-way | Required | Required | Required |
| Two-way | Required | Required | Not Required |
| Loopback | Required | Not Required | Not Required |

### One-Way Measurement Mode

One-way measurement mode provides the most precise form of one-way delay measurement. PTP-capable hardware and hardware timestamping are required on both Sender and Reflector, with PTP Clock Synchronization between Sender and Reflector.

Delay measurement in one-way mode is calculated as (T2 – T1).

*Figure 1: One-Way*




The PM query and response for one-way delay measurement can be described in the following steps:

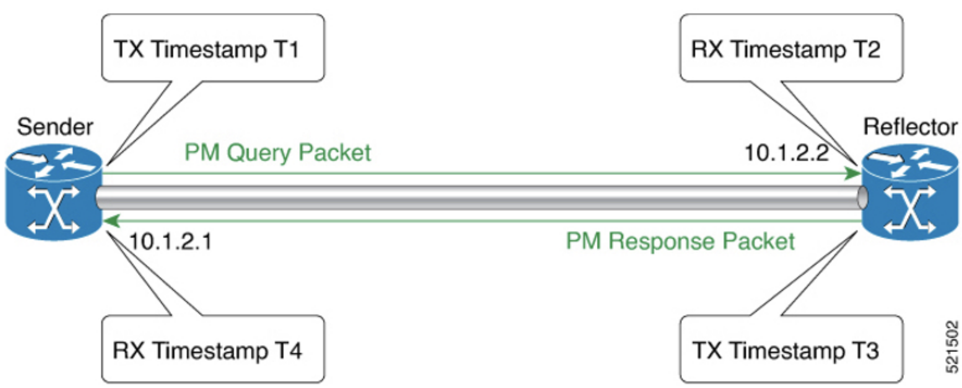
1. The local-end router sends PM query packets periodically to the remote side once the egress line card on the router applies timestamps on packets.
2. The ingress line card on the remote-end router applies time-stamps on packets as soon as they are received.
3. The remote-end router sends the PM packets containing time-stamps back to the local-end router.
4. One-way delay is measured using the time-stamp values in the PM packet.

## Two-Way Measurement Mode

Two-way meaurement mode provides two-way measurements. PTP-capable hardware and hardware timestamping are required on both Sender and Reflector, but PTP clock synchronization between Sender and Reflector is not required.

Delay measurement in two-way mode is calculated as ((T4 – T1) – (T3 – T2))/2.

*Figure 2: Two-Way*




The PM query and response for two-way delay measurement can be described in the following steps:

1. The local-end router sends PM query packets periodically to the remote side once the egress line card on the router applies timestamps on packets.

2. Ingress line card on the remote-end router applies time-stamps on packets as soon as they are received.
3. The remote-end router sends the PM packets containing time-stamps back to the local-end router. The remote-end router time-stamps the packet just before sending it for two-way measurement.
4. The local-end router time-stamps the packet as soon as the packet is received for two-way measurement.
5. Delay is measured using the time-stamp values in the PM packet.

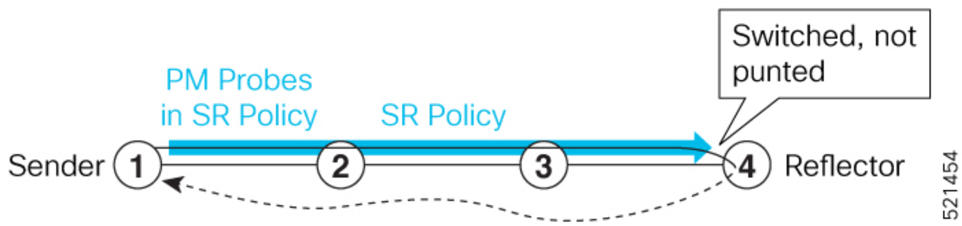
### Loopback Measurement Mode

Loopback meaurement mode provides two-way and one-way measurements. PTP-capable hardware and hardware timestamping are required on the Sender, but are not required on the Reflector.

Delay measurements in Loopback mode are calculated as follows:

- Round-Trip Delay = (T4 – T1)
- One-Way Delay = Round-Trip Delay/2

*Figure 3: Loopback*




The PM query and response for Loopback delay measurement can be described in the following steps:

1. The local-end router sends PM probe packets periodically on the SR Policy.
2. The probe packets are loopback on the endpoint node (not punted), with no timestamping on endpoint node.
3. Round-trip Delay = T4 – T1.

# Usage Guidelines and Limitations

The following usage guidelines and limitations apply:

- SR PM is supported on hardware that supports Precision Time Protocol (PTP). This requirement applies to both one-way and two-way delay measurement.

  See the "**Configuring Precision Time Protocol**" chapter in the *System Management Configuration Guide for Cisco 8000 Series Routers* for Restrictions for PTP and the Timing Hardware Support Matrix.

# Link Delay Measurement

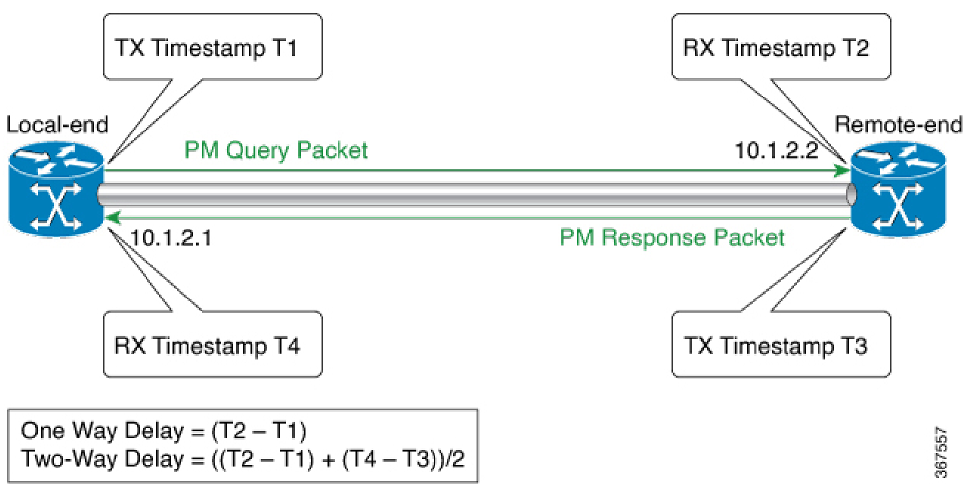
*Table 3: Feature History Table*

| Feature Name | Release Information | Feature Description |
|---|---|---|
| Link Delay Measurement using TWAMP Light Encoding | Release 7.3.1 | The PM for link delay uses the IP/UDP packet format defined in RFC 5357 (TWAMP-Light) for probes. Two-Way Active Measurement Protocol (TWAMP) adds two-way or round-trip measurement capabilities. TWAMP employs time stamps applied at the echo destination (reflector) to enable greater accuracy. |

The PM for link delay uses the IP/UDP packet format defined in RFC 5357 (TWAMP-Light) for probes. Two-Way Active Measurement Protocol (TWAMP) adds two-way or round-trip measurement capabilities. TWAMP employs time stamps applied at the echo destination (reflector) to enable greater accuracy. In the case of TWAMP Light, the Session-Reflector doesn't necessarily know about the session state. The Session-Reflector simply copies the Sequence Number of the received packet to the Sequence Number field of the reflected packet. The controller receives the reflected test packets and collects two-way metrics. This architecture allows for collection of two-way metrics.

The following figure explains the PM query and response for link delay.

*Figure 4: Performance Measurement for Link Delay*




The PM query and response for link delay can be described in the following steps:

1. The local-end router sends PM query packets periodically to the remote side once the egress line card on the router applies timestamps on packets.
2. Ingress line card on the remote-end router applies time-stamps on packets as soon as they are received.

3. The remote-end router sends the PM packets containing time-stamps back to the local-end router. The remote-end router time-stamps the packet just before sending it for two-way measurement.
4. The local-end router time-stamps the packet as soon as the packet is received for two-way measurement.
5. One-way delay and optionally two-way delay is measured using the time-stamp values in the PM packet.

### Restrictions and Usage Guidelines for PM for Link Delay

The following restrictions and guidelines apply for the PM for link delay feature for different links.

- For LSPs, remote-end line card needs to be MPLS and multicast MAC address capable.
- For broadcast links, only point-to-point (P2P) links are supported. P2P configuration on IGP is required for flooding the value.
- For link bundles, the hashing function may select a member link for forwarding but the reply may come from the remote line card on a different member link of the bundle.
- For one-way delay measurement, clocks should be synchronized on two end-point nodes of the link using PTP.
- Link delay measurement is supported on IPv4 unnumbered interfaces. An IPv4 unnumbered interface is identified by a node ID (a loopback address) and the local SNMP index assigned to the interface. Note that the reply messages could be received on any interface, since the packets are routed at the responder based on the loopback address used to identify the link.

### Configuration Example: PM for Link Delay

This example shows how to configure performance-measurement functionalities for link delay as a global default profile. The default values for the different parameters in the PM for link delay is given as follows:

- **probe measurement mode**: The default measurement mode for probe is two-way delay measurement. If you are configuring one-way delay measurement, hardware clocks must be synchronized between the local-end and remote-end routers using precision time protocol (PTP).
- **protocol**: Interface delay measurement uses TWAMP-Light (RFC5357) with IP/UDP encap.
- **burst-interval**: Interval for sending probe packet. The default value is 3000 milliseconds and the range is from 30 to 15000 milliseconds.
- **computation interval**: Interval for metric computation. Default is 30 seconds; range is 1 to 3600 seconds.
- **periodic advertisement**: Periodic advertisement is enabled by default.
- **periodic-advertisement interval**: The default value is 120 seconds and the interval range is from 30 to 3600 seconds.
- **periodic-advertisement threshold**: The default value of periodic advertisement threshold is 10 percent.
- **periodic-advertisement minimum change**: The default value is 1000 microseconds (usec) and the range is from 0 to 10000 microseconds.
- **accelerated advertisement**: Accelerated advertisement is disabled by default.
- **accelerated-advertisement threshold**: The default value is 20 percent and the range is from 0 to 100 percent.

- **accelerated-advertisement minimum change**: The default value is 1000 microseconds and the range is from 1 to 100000 microseconds.

```
RP/0/0/CPU0:router(config)# performance-measurement delay-profile interfaces
RP/0/0/CPU0:router(config-pm-dm-intf)# probe
RP/0/0/CPU0:router(config-pm-dm-intf-probe)# measurement-mode one-way
RP/0/0/CPU0:router(config-pm-dm-intf-probe)# burst-interval 60
RP/0/0/CPU0:router(config-pm-dm-intf-probe)# exit

RP/0/0/CPU0:router(config-pm-dm-intf)# advertisement periodic
RP/0/0/CPU0:router(config-pm-dm-intf-adv-per)# interval 120
RP/0/0/CPU0:router(config-pm-dm-intf-adv-per)# threshold 20
RP/0/0/CPU0:router(config-pm-dm-intf-adv-per)# minimum-change 1000
RP/0/0/CPU0:router(config-pm-dm-intf-adv-per)# exit

RP/0/0/CPU0:router(config-pm-dm-intf)# advertisement accelerated
RP/0/0/CPU0:router(config-pm-dm-intf-adv-acc)# threshold 30
RP/0/0/CPU0:router(config-pm-dm-intf-adv-acc)# minimum-change 1000
RP/0/0/CPU0:router(config-pm-dm-intf-adv-per)# exit
```

### Configure the UDP Destination Port

Configuring the UDP port for TWAMP-Light protocol is optional. By default, PM uses port 862 as the TWAMP-reserved UDP destination port.

The UDP port is configured for each PM measurement probe type (delay, loss, protocol, authentication mode, etc.) on querier and responder nodes. If you configure a different UDP port, the UDP port for each PM measurement probe type must match on the querier and the responder nodes.

**Note** The same UDP destination port is used for delay measurement for links and SR Policy.

This example shows how to configure the UDP destination port.

```
Router(config)# performance-measurement
Router(config-perf-meas)# protocol twamp-light
Router(config-pm-protocol)# measurement delay unauthenticated
Router(config-pm-proto-mode)# querier-dst-port 12000
```

### Enable PM for Link Delay Over an Interface

This example shows how to enable PM for link delay over an interface.

```
RP/0/0/CPU0:router(config)# performance-measurement
RP/0/0/CPU0:router(config-perf-meas)# interface TenGigE0/0/0/0
RP/0/0/CPU0:router(config-pm-intf)# delay-measurement
RP/0/0/CPU0:router(config-pm-dm-intf)# exit
```

### Verification

```
RP/0/0/CPU0:router# show performance-measurement profile interface
Thu Dec 12 14:13:16.029 PST

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

```
0/0/CPU0
--------------------------------------------------------------------------------
Interface Delay-Measurement:
  Profile configuration:
    Measurement Type                                : Two-Way
    Probe computation interval                      : 30 (effective: 30) seconds
    Type of services                                : Traffic Class: 6, DSCP: 48
    Burst interval                                  : 3000 (effective: 3000) mSec
    Burst count                                     : 10 packets
    Encap mode                                      : UDP
    Payload Type                                    : TWAMP-light
    Destination sweeping mode                       : Disabled
    Periodic advertisement                          : Enabled
      Interval                                      : 120 (effective: 120) sec
      Threshold                                     : 10%
      Minimum-Change                                : 500 uSec
    Advertisement accelerated                       : Disabled
    Threshold crossing check                        : Minimum-delay

RP/0/0/CPU0:router# show performance-measurement summary detail location 0/2/CPU0

Thu Dec 12 14:09:59.162 PST

--------------------------------------------------------------------------------
0/2/CPU0
--------------------------------------------------------------------------------
Total interfaces                                    : 1
Total SR Policies                                   : 0
Total RSVP-TE tunnels                               : 0
Total Maximum PPS                                   : 2000 pkts/sec
Total Interfaces PPS                                : 0 pkts/sec
Maximum Allowed Multi-hop PPS                       : 2000 pkts/sec
Multi Hop Requested PPS                             : 0 pkts/sec (0% of max allowed)
Dampened Multi Hop Requested PPS                    : 0% of max allowed
Inuse Burst Interval Adjustment Factor              : 100% of configuration


Interface Delay-Measurement:
  Total active sessions                             : 1
  Counters:
    Packets:
      Total sent                                    : 26
      Total received                                : 26
    Errors:
        TX:
          Reason interface down                     : 0
          Reason no MPLS caps                       : 0
          Reason no IP address                      : 0
          Reason other                              : 0
        RX:
          Reason negative delay                     : 0
          Reason delay threshold exceeded           : 0
          Reason missing TX timestamp               : 0
          Reason missing RX timestamp               : 0
          Reason probe full                         : 0
          Reason probe not started                  : 0
          Reason control code error                 : 0
          Reason control code notif                 : 0
    Probes:
      Total started                                 : 3
      Total completed                               : 2
      Total incomplete                              : 0
      Total advertisements                          : 0
```

```
SR Policy Delay-Measurement:
  Total active sessions                          : 0
  Counters:
    Packets:
      Total sent                                 : 0
      Total received                             : 0
    Errors:
        TX:
          Reason interface down                  : 0
          Reason no MPLS caps                    : 0
          Reason no IP address                   : 0
          Reason other                           : 0
        RX:
          Reason negative delay                  : 0
          Reason delay threshold exceeded        : 0
          Reason missing TX timestamp            : 0
          Reason missing RX timestamp            : 0
          Reason probe full                      : 0
          Reason probe not started               : 0
          Reason control code error              : 0
          Reason control code notif              : 0
    Probes:
      Total started                              : 0
      Total completed                            : 0
      Total incomplete                           : 0
      Total advertisements                       : 0

RSVP-TE Delay-Measurement:
  Total active sessions                          : 0
  Counters:
    Packets:
      Total sent                                 : 0
      Total received                             : 0
    Errors:
        TX:
          Reason interface down                  : 0
          Reason no MPLS caps                    : 0
          Reason no IP address                   : 0
          Reason other                           : 0
        RX:
          Reason negative delay                  : 0
          Reason delay threshold exceeded        : 0
          Reason missing TX timestamp            : 0
          Reason missing RX timestamp            : 0
          Reason probe full                      : 0
          Reason probe not started               : 0
          Reason control code error              : 0
          Reason control code notif              : 0
    Probes:
      Total started                              : 0
      Total completed                            : 0
      Total incomplete                           : 0
      Total advertisements                       : 0

Global Delay Counters:
  Total packets sent                             : 26
  Total query packets received                   : 26
  Total invalid session id                       : 0
  Total missing session                          : 0

RP/0/0/CPU0:router# show performance-measurement interfaces detail
Thu Dec 12 14:16:09.692 PST

--------------------------------------------------------------------------------
0/0/CPU0
```

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

--------------------------------------------------------------------------------
0/2/CPU0
--------------------------------------------------------------------------------

Interface Name: GigabitEthernet0/2/0/0 (ifh: 0x1004060)
  Delay-Measurement                 : Enabled
  Loss-Measurement                  : Disabled
  Configured IPv4 Address           : 10.10.10.2
  Configured IPv6 Address           : 10:10:10::2
  Link Local IPv6 Address           : fe80::3a:6fff:fec9:cd6b
  Configured Next-hop Address       : Unknown
  Local MAC Address                 : 023a.6fc9.cd6b
  Next-hop MAC Address              : 0291.e460.6707
  Primary VLAN Tag                  : None
  Secondary VLAN Tag                : None
  State                             : Up

  Delay Measurement session:
    Session ID          : 1

    Last advertisement:
      Advertised at: Dec 12 2019 14:10:43.138 (326.782 seconds ago)
      Advertised reason: First advertisement
      Advertised delays (uSec): avg: 839, min: 587, max: 8209, variance: 297

    Next advertisement:
      Threshold check scheduled in 1 more probe (roughly every 120 seconds)
      Aggregated delays (uSec): avg: 751, min: 589, max: 905, variance: 112
      Rolling average (uSec): 756

    Current Probe:
      Started at Dec 12 2019 14:15:43.154 (26.766 seconds ago)
      Packets Sent: 9, received: 9
      Measured delays (uSec): avg: 795, min: 631, max: 1199, variance: 164
      Next probe scheduled at Dec 12 2019 14:16:13.132 (in 3.212 seconds)
      Next burst packet will be sent in 0.212 seconds
      Burst packet sent every 3.0 seconds
      Probe samples:
        Packet Rx Timestamp       Measured Delay (nsec)
        Dec 12 2019 14:15:43.156          689223
        Dec 12 2019 14:15:46.156          876561
        Dec 12 2019 14:15:49.156          913548
        Dec 12 2019 14:15:52.157         1199620
        Dec 12 2019 14:15:55.156          794008
        Dec 12 2019 14:15:58.156          631437
        Dec 12 2019 14:16:01.157          656440
        Dec 12 2019 14:16:04.157          658267
        Dec 12 2019 14:16:07.157          736880
```

You can also use the following commands for verifying the PM for link delay on the local-end router.

| Command | Description |
|---|---|
| **show performance-measurement history probe interfaces** [*interface*] | Displays the PM link-delay probe history for interfaces. |
| **show performance-measurement history aggregated interfaces** [*interface*] | Displays the PM link-delay aggregated history for interfaces. |
| **show performance-measurement history advertisement interfaces** [*interface*] | Displays the PM link-delay advertisement history for interfaces. |

| Command | Description |
|---|---|
| **show performance-measurement counters** [interface *interface*] [location *location-name*] | Displays the PM link-delay session counters. |

You can also use the following commands for verifying the PM for link-delay configuration on the remote-end router.

| Command | Description |
|---|---|
| **show performance-measurement responder summary** [**location** *location-name*] | Displays the PM for link-delay summary on the remote-end router (responder). |
| **show performance-measurement responder interfaces** [*interface*] | Displays PM for link-delay for interfaces on the remote-end router. |
| **show performance-measurement responder counters** [**interface** *interface*] [**location** *location-name*] | Displays the PM link-delay session counters on the remote-end router. |

### SR Performance Measurement Named Profiles

| Feature | Release Information | Feature Description |
|---|---|---|
| SR Performance Measurement Named Profiles | Release 7.5.2 | You can use this feature to create specific performance measurement delay and liveness profiles, and associate it with an SR policy.<br>You can use the delay or liveness profile to be associated with an interface or network area, where the performance management probes are enabled, and performance measurement is precise and enhanced. |

You can create a named performance measurement profile for delay or liveness.

**Delay Profile**

This example shows how to create a named SR performance measurement delay profile.

```
Router(config)# performance-measurement delay-profile sr-policy profile2
Router(config-pm-dm-srpolicy)# probe
Router(config-pm-dm-srpolicy-probe)# burst-interval 60
Router(config-pm-dm-srpolicy-probe)# computation-interval 60
Router(config-pm-dm-srpolicy-probe)# protocol twamp-light
Router(config-pm-dm-srpolicy-probe)# tos dscp 63

Router(config-pm-dm-srpolicy)# advertisement
Router(config-pm-dm-srpolicy-adv)# periodic
Router(config-pm-dm-srpolicy-adv-per)# interval 60
Router(config-pm-dm-srpolicy-adv-per)# minimum-change 1000
Router(config-pm-dm-srpolicy-adv-per)# threshold 20
Router(config-pm-dm-srpolicy-adv-per)# commit
```

Apply the delay profile for an SR Policy.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# policy TEST
Router(config-sr-te-policy)# color 4 end-point ipv4 10.10.10.10
Router(config-sr-te-policy)# performance-measurement
Router(config-sr-te-policy-perf-meas)# delay-measurement delay-profile name profile2

Router(config-sr-te-policy)#candidate-paths
Router(config-sr-te-policy-path)#preference 100
Router(config-sr-te-policy-path-pref)#explicit segment-list LIST1
Router(config-sr-te-pp-info)#weight 2

Router(config-sr-te-policy-path-pref)#explicit segment-list LIST2
Router(config-sr-te-pp-info)#weight 3
```

### Running Configuration

```
Router# show run segment-routing traffic-eng policy TEST

segment-routing
 traffic-eng
  policy TEST
   color 4 end-point ipv4 10.10.10.10
   candidate-paths
    preference 100
     explicit segment-list LIST1
      weight 2
     !
     explicit segment-list LIST2
      weight 3
     !
    !
   !
   performance-measurement
    delay-measurement
     delay-profile name profile2
```

### Verification

```
Router# show performance-measurement profile named-profile delay sr-policy name profile2

----------------------------
0/RSP0/CPU0
----------------------------
SR Policy Delay Measurement Profile Name: profile2
  Profile configuration:
    Measurement mode                             : One-way
    Protocol type                                : TWAMP-light
    Encap mode                                   : UDP
    Type of service:
      PM-MPLS traffic class                      : 6
      TWAMP-light DSCP                           : 63
    Probe computation interval                   : 60 (effective: 60) seconds
    Burst interval                               : 60 (effective: 60) mSec
    Packets per computation interval             : 1000
    Periodic advertisement                       : Enabled
      Interval                                   : 60 (effective: 60) sec
      Threshold                                  : 20%
      Minimum-change                             : 1000 uSec
    Advertisement accelerated                    : Disabled
    Advertisement logging:
      Delay exceeded                             : Disabled (default)
    Threshold crossing check                     : Maximum-delay
    Router alert                                 : Disabled (default)
```

```
    Destination sweeping mode                    : Disabled
    Liveness detection parameters:
      Multiplier                                 : 3
      Logging state change                       : Disabled
```

**On-Demand SR Policy**

```
Router(config-sr-te)# on-demand color 20
Router(config-sr-te-color)# performance-measurement delay-measurement
Router(config-sr-te-color-delay-meas)# delay-profile name profile2
Router(config-sr-te-color-delay-meas)# commit
```

**Running Configuration**

```
Router# show run segment-routing traffic-eng on-demand color 20

segment-routing
 traffic-eng
  on-demand color 20
   performance-measurement
    delay-measurement
     delay-profile name profile2
```

### Liveness Profile

This example shows how to create a *named* SR performance measurement liveness profile.

```
Router(config)# performance-measurement liveness-profile sr-policy name profile3
Router(config-pm-ld-srpolicy)# probe
Router(config-pm-ld-srpolicy-probe)# burst-interval 60
Router(config-pm-ld-srpolicy-probe)# measurement-mode loopback
Router(config-pm-ld-srpolicy-probe)# tos dscp 10
Router(config-pm-ld-srpolicy-probe)# liveness-detection
Router(config-pm-ld-srpolicy-probe)# multiplier 5
Router(config-pm-ld-srpolicy-probe)# commit
```

**Apply the liveness profile for the SR policy**

This example shows how to enable PM for SR policy liveness for a specific policy.

For the same policy, you cannot enable delay-measurement (delay-profile) and liveness-detection (liveness-profile) at the same time. For example, if delay measurement is enabled, use the **no delay-measurement** command to disable it, and then enable the following command for enabling liveness detection.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# policy TRST2
Router(config-sr-te-policy)# color 40 end-point ipv4 20.20.20.20
Router(config-sr-te-policy)#candidate-paths
Router(config-sr-te-policy-path)#preference 50
Router(config-sr-te-policy-path-pref)#explicit segment-list LIST3
Router(config-sr-te-pp-info)#weight 2

Router(config-sr-te-policy-path-pref)#explicit segment-list LIST4
Router(config-sr-te-pp-info)#weight 3

Router(config-sr-te-policy)# performance-measurement
Router(config-sr-te-policy-perf-meas)# liveness-detection liveness-profile name profile3
```

**Running Configuration**

```
Router# show run segment-routing traffic-eng policy TRST2

segment-routing
 traffic-eng
```

```
  policy TRST2
   color 40 end-point ipv4 20.20.20.20
   candidate-paths
    preference 50
     explicit segment-list LIST3
      weight 2
     !
     explicit segment-list LIST4
      weight 3
     !
    !
   !
   performance-measurement
    liveness-detection
     liveness-profile name profile3
    !
```

### Verification

```
Router# show performance-measurement profile named-profile delay sr-policy

-----------
0/RSP0/CPU0
-----------

SR Policy Liveness Detection Profile Name: profile1
  Profile configuration:
    Measurement mode                            : Loopback
    Protocol type                               : TWAMP-light
    Type of service:
      TWAMP-light DSCP                          : 10
    Burst interval                              : 60 (effective: 60) mSec
    Destination sweeping mode                   : Disabled
    Liveness detection parameters:
      Multiplier                                : 3
      Logging state change                      : Disabled

SR Policy Liveness Detection Profile Name: profile3
  Profile configuration:
    Measurement mode                            : Loopback
    Protocol type                               : TWAMP-light
    Type of service:
      TWAMP-light DSCP                          : 10
    Burst interval                              : 60 (effective: 60) mSec
    Destination sweeping mode                   : Disabled
    Liveness detection parameters:
      Multiplier                                : 3
      Logging state change                      : Disabled
```

### On-Demand SR Policy

For the same policy, you cannot enable delay-measurement (delay-profile) and liveness-detection (liveness-profile) at the same time. For example, to disable delay measurement, use the **no delay-measurement** command, and then enable the following command for enabling liveness detection.

```
Router(config-sr-te)#on-demand color 30
Router(config-sr-te-color)#performance-measurement
Router(config-sr-te-color-pm)# liveness-detection liveness-profile name profile1
Router(config-sr-te-color-delay-meas)# commit
```

### Running Configuration

```
Router# show run segment-routing traffic-eng on-demand color 30

segment-routing
```

```
 traffic-eng
  on-demand color 30
   performance-measurement
    liveness-detection
     liveness-profile name profile1
    !
```

**Verification**

```
Router# show performance-measurement profile named-profile liveness sr-policy name profile1

--------------------
0/RSP0/CPU0
--------------------
SR Policy Liveness Detection Profile Name: profile1
  Profile configuration:
    Measurement mode                               : Loopback
    Protocol type                                  : TWAMP-light
    Type of service:
      TWAMP-light DSCP                             : 10
    Burst interval                                 : 60 (effective: 60) mSec
    Destination sweeping mode                      : Disabled
    Liveness detection parameters:
      Multiplier                                   : 3
      Logging state change                         : Disabled
```

# Delay Normalization

Performance measurement (PM) measures various link characteristics like packet loss and delay. Such characteristics can be used by IS-IS as a metric for Flexible Algorithm computation. Low latency routing using dynamic delay measurement is one of the primary use cases for Flexible Algorithm technology.

Delay is measured in microseconds. If delay values are taken as measured and used as link metrics during the IS-IS topology computation, some valid ECMP paths might be unused because of the negligible difference in the link delay.

The Delay Normalization feature computes a normalized delay value and uses the normalized value instead. This value is advertised and used as a metric during the Flexible Algorithm computation.

The normalization is performed when the delay is received from the delay measurement component. When the next value is received, it is normalized and compared to the previous saved normalized value. If the values are different, then the LSP generation is triggered.

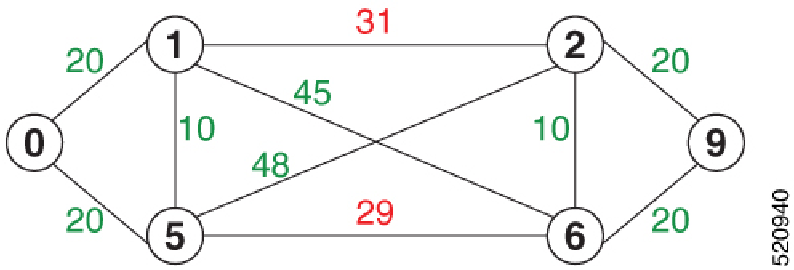
The following formula is used to calculate the normalized value:

- **Dm** – measured Delay
- **Int** – configured normalized Interval
- **Off** – configured normalized Offset (must be less than the normalized interval Int)
- **Dn** – normalized Delay
- **a** = Dm / Int (rounded down)
- **b** = a * Int + Off

If the measured delay (Dm) is less than or equal to **b**, then the normalized delay (Dn) is equal to **b**. Otherwise, Dn is **b + Int**.

### Example

The following example shows a low-latency service. The intent is to avoid high-latency links (1-6, 5-2). Links 1-2 and 5-6 are both low-latency links. The measured latency is not equal, but the difference is insignificant.




We can normalize the measured latency before it is advertised and used by IS-IS. Consider a scenario with the following:

- Interval = 10
- Offset = 3

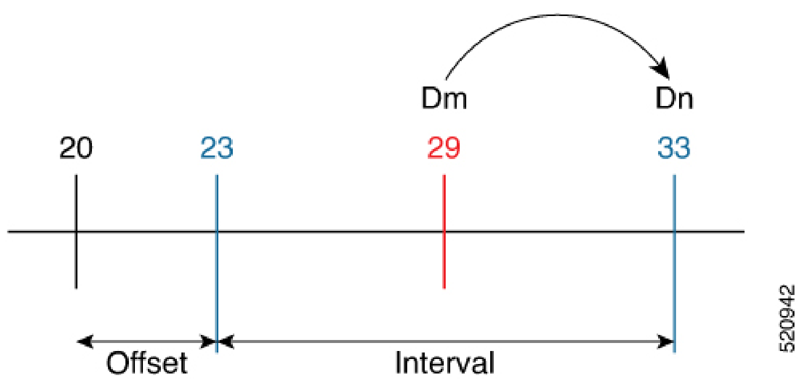
The measured delays will be normalized as follows:

- **Dm** = 29

  **a** = 29 / 10 = 2 (2.9, rounded down to 2)

  **b** = 2 * 10 + 3 = 23

  In this case, **Dm** (29) is greater than **b** (23); so **Dn** is equal to **b+I** (23 + 10) = **33**

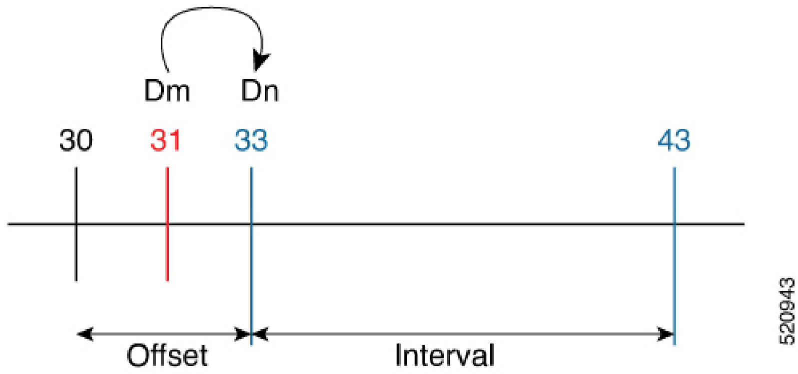



- **Dm** = 31

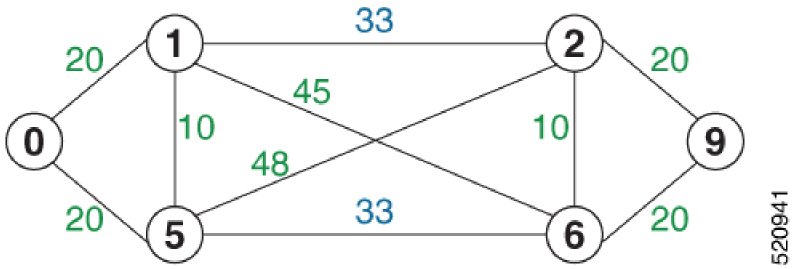
  **a** = 31 / 10 = 3 (3.1, rounded down to 3)

  **b** = 3 * 10 + 3 = 33

  In this case, **Dm** (31) is less than **b** (33); so **Dn** is **b** = **33**




The link delay between 1-2 and 5-6 is normalized to 33.




### Configuration

Delay normalization is disabled by default. To enable and configure delay normalization, use the **delay normalize interval** *interval* [**offset** *offset*] command.

- *interval* – The value of the normalize interval in microseconds.
- *offset* – The value of the normalized offset in microseconds. This valude must be smaller than the value of normalized interval.

#### IS-IS Configuration

```
router isis 1
 interface GigEth 0/0/0/0
  delay normalize interval 10 offset 3
  address-family ipv4 unicast
   metric 77
```

#### OSPF Configuration

```
router ospf 1
 area 0
  interface GigabitEthernet0/0/0/0
   delay normalize interval 10 offset 3
  !
 !
!
```

# Link Anomaly Detection with IGP Penalty

***Table 4: Feature History Table***

| Feature Name | Release Information | Feature Description |
|---|---|---|
| **Link Anomaly Detection with IGP Penalty** | Release 7.4.1 | This feature allows you to define thresholds above the measured delay that is considered “anomalous” or unusual. When this threshold is exceeded, an anomaly (A) bit/flag is set along with link delay attribute that is sent to clients. |

Customers might experience performance degradation issues, such as increased latency or packet loss on a link. Degraded links might be difficult to troubleshoot and can affect applications, especially in cases where traffic is sent over multiple ECMP paths where one of those paths is degraded.

The Anomaly Detection feature allows you to define a delay anomaly threshold to identify unacceptable link delays. Nodes monitor link performance using link delay monitoring probes. The measured value is compared against the delay anomaly threshold values. When the upper bound threshold is exceeded, the link is declared “abnormal”, and performance measurement sets an anomaly bit (A-bit). When IGP receives the A-bit, IGP can automatically increase the IGP metric of the link by a user-defined amount to make this link undesirable or unusable. When the link recovers (lower bound threshold), PM resets the A-bit.

### Usage Guidelines and Limitations

This feature is not active when narrow metrics are configured because the performance measurement advertisement requires the “wide” metric type length values.

### Configuration Example

The following example shows how to configure the upper and lower anomoly thresholds. The range for *upper_bound* and *lower_bound* is from 1 to 200,000 microseconds. The *lower_bound* value must be less than the *upper_bound* value.

```
RP/0/0/CPU0:router(config)# performance-measurement delay-profile interfaces default
RP/0/0/CPU0:router(config-pm-dm-intf)# advertisement
RP/0/0/CPU0:router(config-pm-dm-intf-adv)# anomaly-check upper-bound 5000 lower-bound 1000
RP/0/0/CPU0:router(config-pm-dm-intf-adv)# commit
```

### Running Configuration

```
performance-measurement
 delay-profile interfaces default
  advertisement
   anomaly-check
    upper-bound 5000 lower-bound 1000
   !
  !
 !
!
end
```

# IP Endpoint Delay Measurement and Liveness Monitoring

***Table 5: Feature History Table***

| Feature Name | Release Information | Feature Description |
|---|---|---|
| IP Endpoint Delay Measurement and Liveness Monitoring | Release 7.4.1 | This feature measures the end-to-end delay and monitors liveness of a specified IP endpoint node, including VRF-aware (awareness of multiple customers belonging to different VRFs).<br>This feature is supported on IPv4, IPv6, and MPLS data planes. |

The Segment Routing Performance Measurement for IP Endpoint feature dynamically measures the end-to-end delay and liveness towards a specified IP endpoint. IP endpoints can be located in default or non-default VRFs.

The following are benefits of this feature:

- Performance values (delay metrics and liveness states) are computed using TWAMP light, which is a commonly supported protocol.
- Performance values, including Histograms, are sent out using Streaming Telemetry, which is a push-based data collection technique, rather than a manual data collection technique.
- The feature is applicable to multiple data plane types (e.g. IPv4, IPv6, and MPLS).

## Configuring IP Endpoint Performance Measurement

### Usage Guidelines and Limitations

Observe the following usage guidelines and limitations:

- The endpoint of a probe is specified with an IP address. IPv4 and IPv6 endpoint addresses are supported.
- The endpoint of a probe can be any IP address reachable by the sender. For example, a local interface or a remote node or host located within an operator's network or reachable through a VRF.
- The endpoint's IP address can be located in the global routing table or under a user-specified VRF routing table.
- VRF-awareness allows operators to deploy probes in the following scenarios:
    - Managed Customer Equipment (CE) scenarios:
        - PE to CE probes
        - CE to CE probes
    - Unmanaged Customer Equipment (CE) scenarios:
        - PE to PE probes

- PE to PE (source from PE-CE interface) probes

- SRv6 locator prefix and VRF SRv6 locator/function (uDT4/uDT6) as IPv6 endpoint of a probe is not supported.
- The endpoint's IP address can be reached through an IP path, MPLS LSP, or IP tunnel (GRE).
- When the endpoint is reachable using an MPLS LSP (for example, SR, LDP, RSVP-TE, SR Policy), the forwarding stage imposes the corresponding MPLS transport labels.
- When the endpoint is reachable via a GRE tunnel, the forwarding stage imposes the corresponding GRE header.
- PM probe over GREv4 is supported.
- When the endpoint is reachable via a VRF in an MPLS network, the forwarding stage imposes the corresponding MPLS service labels. In the forward path, sender node uses the configured VRF for the endpoint address. In the return path, reflector node derives the VRF based on which incoming VRF label the probe packet is received with.

### Configuring Probe Endpoint Source and Destination Addresses

Use the **performance-measurement endpoint** {**ipv4** | **ipv6**} *ip_addr* [**vrf** *WORD*] **source-address** {**ipv4** | **ipv6**} *ip_addr* command to configure a probe endpoint source and destination addresses on a sender node.

**Example:**

```
Router(config)# performance-measurement endpoint ipv4 1.1.1.5 source-address ipv4 1.1.1.1

Router(config)# performance-measurement endpoint ipv4 10.10.10.100 vrf green source-address
 ipv4 1.1.1.1
```

### Configuring Probe Description

Use the **performance-measurement endpoint** {**ipv4** | **ipv6**} *ip_addr* [**vrf** *WORD*] **description** *LINE* command to configure a probe description.

**Example:**

```
Router(config)# performance-measurement endpoint ipv4 1.1.1.5 description Probe to 1.1.1.5

Router(config)# performance-measurement endpoint ipv4 10.10.10.100 vrf green description
Probe to 10.10.10.100
```

### Configuring Probe Segment-lists

Observe the following usage guidelines and limitations:

- You can specify a custom labeled path via one or more user-configured segment-list. User-configured segment-list represents the forwarding path from sender to reflector when probe configured in delay-measurement mode.
- User-configured segment-list can also represent the reverse path (reflector to sender) when probe configured in livenes-detection mode.

- Up to 128 segment-lists can be configured under a probe.
- An additional PM session is created for each segment-list.
- Examples of the custom segment-list include:
  - Probe in delay-measurement mode with segment-list that includes Flex-Algo prefix SID of the end-point
  - Probe in liveness-detection mode with segment-list that includes both Flex-Algo prefix SID of the end-point and the sender
  - Probe in delay-measurement mode with segment-list that includes SID-list with labels to reach the end-point or the sender (forward direction)
  - Probe in liveness-detection mode with segment-list that includes SID-list with labels to reach the end-point and then back to sender (forward and reverse directions, respectively)
  - Probe in delay-measurement mode with segment-list that includes BSID associated with SR policy to reach the end-point
- Endpoint segment list configuration not supported under non-default VRF.

Segment-lists are configured under **segment-routing traffic-eng segment-list** submode. See SR-TE Policy with Explicit Path for details about configuring segment lists.

Use the **performance-measurement endpoint** {**ipv4** | **ipv6**} *ip_addr* [**vrf** *WORD*] **segment-list name** *WORD* command to configure probe segment-lists.

**Example:**

```
Router(config)# performance-measurement endpoint ipv4 1.1.1.5 segment-list name SIDLIST1

Router(config)# performance-measurement endpoint ipv4 10.10.10.100 vrf green segment-list
name SIDLIST1
```

### Enabling Delay Measurement

Observe the following usage guidelines and limitations:

- Probe dynamically measures end-to-end performance delay values of an IP endpoint.
- Probe can be configured in either liveness-monitoring or delay-measurement modes; not both concurrently.

Use the **performance-measurement endpoint** {**ipv4** | **ipv6**} *ip_addr* [**vrf** *WORD*] **delay-measurement** command to enable delay measurement.

**Example:**

```
Router(config)# performance-measurement endpoint ipv4 1.1.1.5 delay-measurement

Router(config)# performance-measurement endpoint ipv4 10.10.10.100 vrf green delay-measurement
```

### Assigning a Delay-Profile to the Probe

Use the **performance-measurement endpoint** {**ipv4** | **ipv6**} *ip_addr* [**vrf** *WORD*] **delay-measurement delay-profile name** *WORD* command to assign a delay-profile associated with the probe.

**Example:**

```
Router(config)# performance-measurement endpoint ipv4 1.1.1.5 delay-measurement delay-profile
 name PROFILE1


Router(config)# performance-measurement endpoint ipv4 10.10.10.100 vrf green delay-measurement
 delay-profile name PROFILE1
```

### Enabling Liveness Detection

Observe the following usage guidelines and limitations:

- Probe dynamically monitors liveness of an IP endpoint.
- Liveness monitoring uses "self-addressed" PM IP packets crafted by the sender (where IP DA = sender's own IP address); this mode of operation is referred as "loopback mode".
- Liveness monitoring applies to endpoints reachable through MPLS LSP or IP tunnel.
- Liveness monitoring does not apply to endpoints reachable through IP path since sender's self-addressed PM IP packets would not be able to reach the intended endpoint destination.
- Probe can be configured in either liveness-monitoring or delay-measurement modes; not both concurrently.

Use the **performance-measurement endpoint** {**ipv4** | **ipv6**} *ip_addr* [**vrf** *WORD*] **liveness-detection** command to enable liveness detection.

**Example:**

```
Router(config)# performance-measurement endpoint ipv4 1.1.1.5 liveness-detection


Router(config)# performance-measurement endpoint ipv4 10.10.10.100 vrf green
liveness-detection
```

### Assigning Liveness-Profile to the Probe

Use the **performance-measurement endpoint** {**ipv4** | **ipv6**} *ip_addr* [**vrf** *WORD*] **liveness-detection liveness-profile name** *WORD* command to configure the liveness-profile associated with the probe.

**Example:**

```
Router(config)# performance-measurement endpoint ipv4 1.1.1.5 liveness-detection
liveness-profile name PROFILE3


Router(config)# performance-measurement endpoint ipv4 10.10.10.100 vrf green
liveness-detection liveness-profile name PROFILE3
```

### Collecting IP Endpoint Probe Statistics

- Statistics associated with the probe (computed delay metrics and liveness state) are available via Histogram and Streaming Telemetry.
- Model Driven Telemetry (MDT) is supported for the following data:
    - Summary, endpoint, session, and counter show command bags

    - History buffers data

- Model Driven Telemetry (MDT) and Event Driven Telemetry (EDT) are supported for the following data:
    - Delay metrics computed in the last probe computation-interval (event: probe-completed)
    - Delay metrics computed in the last aggregation-interval; i.e. end of the periodic advertisement-interval (event: advertisement-interval expired)
    - Delay metrics last notified (event: notification-triggered)

- The following xpaths for MDT/EDT are supported:
    - Cisco-IOS-XR-perf-meas-oper:performance-measurement/nodes/node/endpoints/endpoint-delay/endpoint-last-probes
    - Cisco-IOS-XR-perf-meas-oper:performance-measurement/nodes/node/endpoints/endpoint-delay/endpoint-last-aggregations
    - Cisco-IOS-XR-perf-meas-oper:performance-measurement/nodes/node/endpoints/endpoint-delay/endpoint-last-advertisements

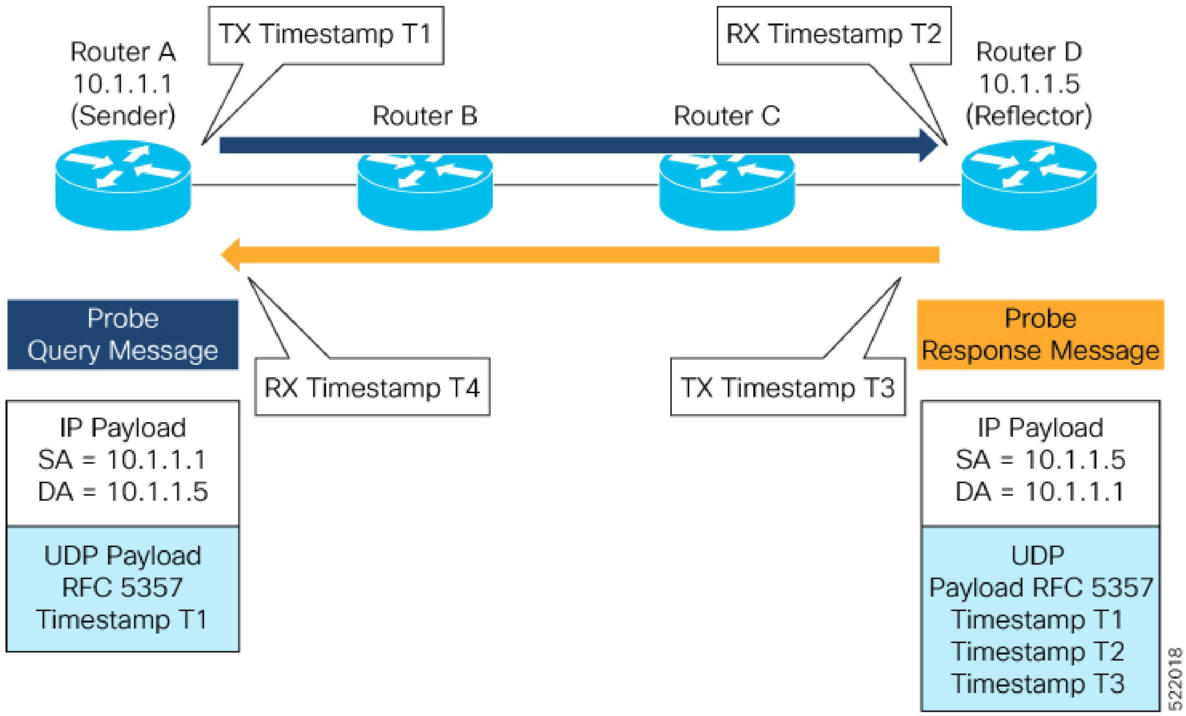
# IP Endpoint Performance Measurement Use Cases

The following use-cases show different ways to deploy delay measurement and liveness detection for IP endpoints.

### Use-Case 1: Delay Measurement Probe Toward an IP Endpoint Reachable in the Global Routing Table

The following figure illustrates a delay measurement probe toward an IP endpoint reachable in the global routing table. The network interconnecting the sender and the reflector provides plain IP connectivity.

*Figure 5: Delay Measurement Probe Toward an IP Endpoint Reachable in the Global Routing Table*




### Configuration

```
RouterA(config)# performance-measurement
RouterA(config-perf-meas)# endpoint ipv4 1.1.1.5
RouterA(config-pm-ep)# source-address ipv4 1.1.1.1
RouterA(config-pm-ep)# delay-measurement
RouterA(config-pm-ep-dm)# exit
RouterA(config-pm-ep)# exit
RouterA(config-perf-meas)# delay-profile endpoint default
RouterA(config-pm-dm-ep)# probe
RouterA(config-pm-dm-ep-probe)# measurement-mode one-way
```

### Running Configuration

```
performance-measurement
 endpoint ipv4 1.1.1.5
  source-address ipv4 1.1.1.1
  delay-measurement
  !
 !
 delay-profile endpoint default
  probe
   measurement-mode one-way
  !
 !
!
```

### Verification

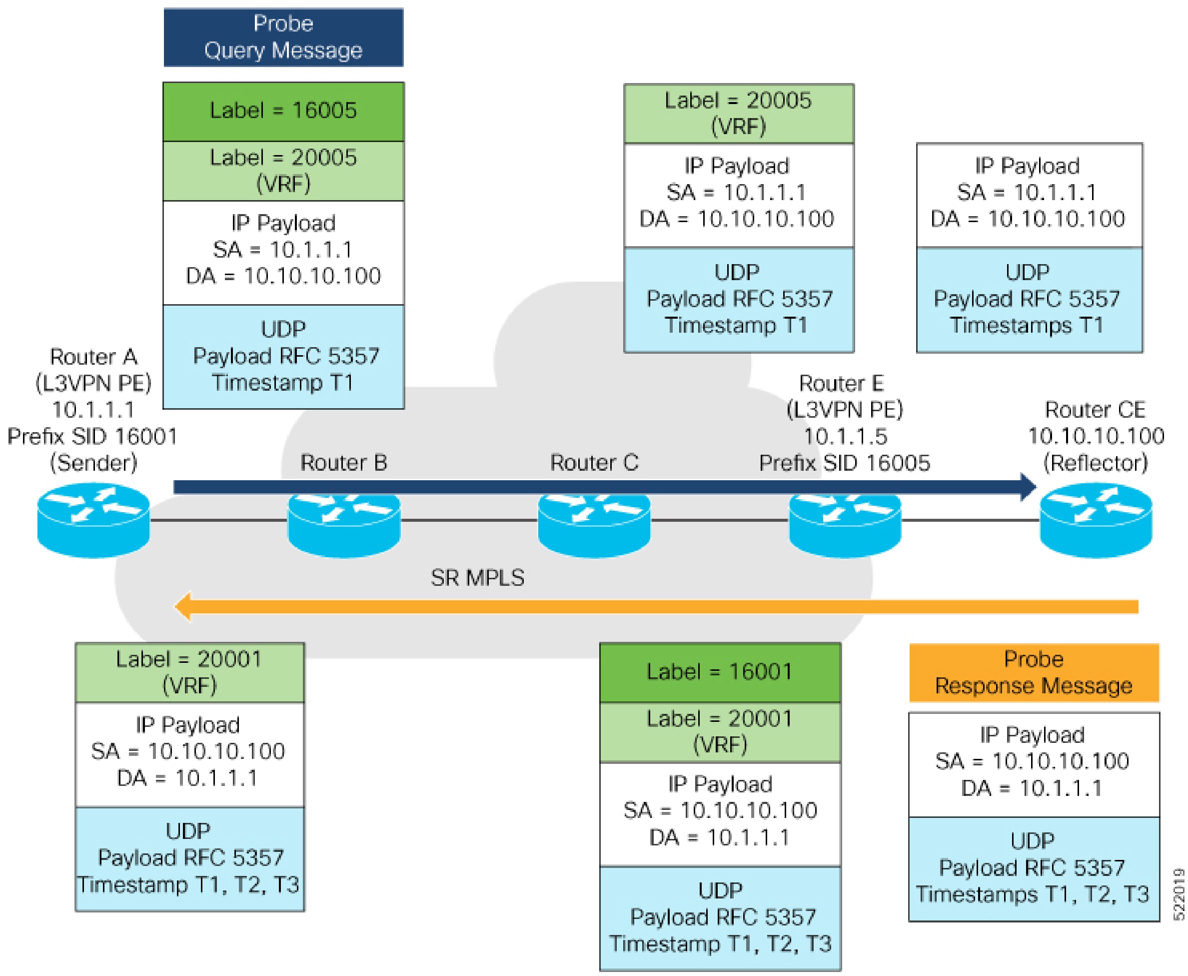
```
RouterA# show performance-measurement endpoint ipv4 1.1.1.5
```

```
0/RSP0/CPU0
--------------------------------------------------------------------------------

Endpoint name: IPv4-1.1.1.5-vrf-default
  Source address             : 1.1.1.1
  VRF name                   : default
  Delay-measurement          : Enabled
  Description                : Not set
  Profile Keys:
    Profile name             : default
    Profile type             : Endpoint Delay Measurement


  Segment-list               : None
  Delay Measurement session:
    Session ID          : 33554433
    Last advertisement:
      No advertisements have occured

    Next advertisement:
      Threshold check scheduled in 4 more probes (roughly every 120 seconds)
      No probes completed

    Current computation:
      Started at: Jul 19 2021 16:28:06.723 (17.788 seconds ago)
      Packets Sent: 6, received: 0
      Measured delays (uSec): avg: 0, min: 0, max: 0, variance: 0
      Next probe scheduled at: Jul 19 2021 16:28:36.718 (in 12.207 seconds)
      Next burst packet will be sent in 0.207 seconds
      Burst packet sent every 3.0 seconds
```

### Use-Case 2: Delay Measurement Probe Toward an IP Endpoint Reachable in a User-Specified VRF

The following figure illustrates a delay measurement probe toward an IP endpoint reachable in a user-specified L3VPN's VRF routing table. The L3VPN ingress PE (Router A) acts as the sender. The reflector is located in a CE device behind the L3VPN egress PE (Router E). The network interconnecting the L3VPN PEs provides MPLS connectivity with Segment Routing.

*Figure 6: Delay Measurement Probe Toward an IP Endpoint Reachable in a User-Specified VRF*




## Configuration

```
RouterA(config)# performance-measurement
RouterA(config-perf-meas)# endpoint ipv4 10.10.10.100 vrf green
RouterA(config-pm-ep)# source-address ipv4 1.1.1.1
RouterA(config-pm-ep)# delay-measurement
RouterA(config-pm-ep-dm)# exit
RouterA(config-pm-ep)# exit
RouterA(config-perf-meas)# delay-profile endpoint default
RouterA(config-pm-dm-ep)# probe
RouterA(config-pm-dm-ep-probe)# measurement-mode one-way
```

## Running Configuration

```
performance-measurement
 endpoint ipv4 10.10.10.100 vrf green
  source-address ipv4 1.1.1.1
  delay-measurement
  !
 !
 delay-profile endpoint default
  probe
   measurement-mode one-way
```

```
    !
  !
!
```

## Verification

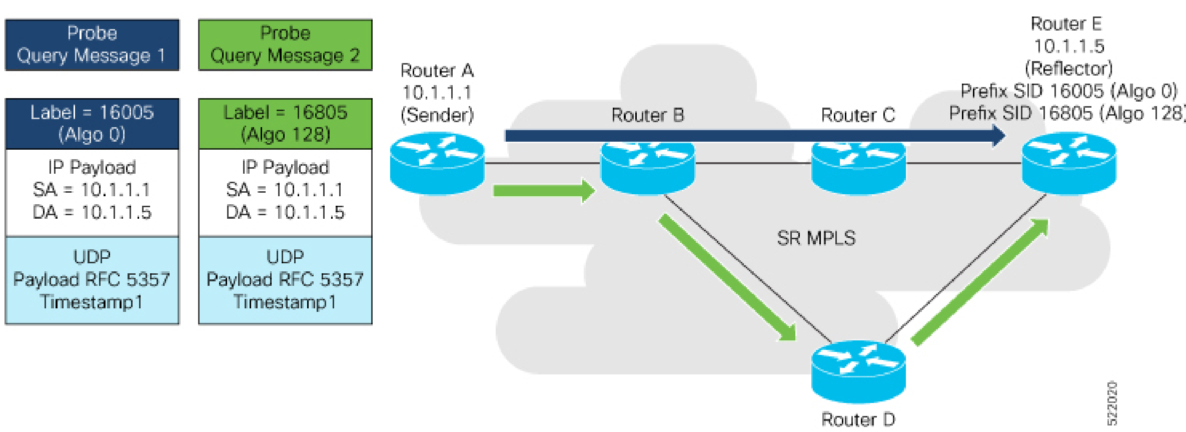
```
RouterA# show performance-measurement endpoint vrf green
---------------------------------------------------------------------------------------------
0/RSP0/CPU0
---------------------------------------------------------------------------------------------

Endpoint name: IPv4-10.10.10.100-vrf-green
  Source address              : 1.1.1.1
  VRF name                    : green
  Delay-measurement           : Enabled
  Description                 : Not set
  Profile Keys:
    Profile name              : default
    Profile type              : Endpoint Delay Measurement


  Segment-list                : None
  Delay Measurement session:
    Session ID          : 33554434
    Last advertisement:
      No advertisements have occured

    Next advertisement:
      Advertisement not scheduled as the probe is not running

    Current computation:
      Not running: Unable to resolve (non-existing) vrf
```

### Use Case 3: Delay Measurement Probe Toward an IP Endpoint Using Custom Labeled Paths

The following figure illustrates a delay measurement probe toward an IP endpoint learned by the IGP. The network interconnecting the sender and reflector provides MPLS connectivity with Segment Routing.

The IP endpoint is advertised with multiple SR algorithms (Algo 0 and Flex Algo 128). The probe is configured with two custom-labeled paths in order to monitor the LSP for each algorithm separately.

*Figure 7: Delay Measurement Probe Toward an IP Endpoint Using Custom Labeled Paths*



**Note** The probe response messages are not shown in the above figure.

### Configuration

```
RouterA(config)# segment-routing
RouterA(config-sr)# traffic-eng
RouterA(config-sr-te)# segment-list name SIDLIST1-Algo0
RouterA(config-sr-te-sl)# index 10 mpls label 16005
RouterA(config-sr-te-sl)# exit

RouterA(config-sr-te)# segment-list name SIDLIST2-FlexAlgo128
RouterA(config-sr-te-sl)# index 10 mpls label 16085
RouterA(config-sr-te-sl)# exit
RouterA(config-sr-te)# exit
RouterA(config-sr)# exit

RouterA(config)# performance-measurement
RouterA(config-perf-meas)# endpoint ipv4 1.1.1.5
RouterA(config-pm-ep)# source-address ipv4 1.1.1.1
RouterA(config-pm-ep)# segment-list name SIDLIST1-Algo0
RouterA(config-pm-ep-sl)# exit
RouterA(config-pm-ep)# segment-list name SIDLIST2-FlexAlgo128
RouterA(config-pm-ep-sl)# exit
RouterA(config-pm-ep)# delay-measurement
RouterA(config-pm-ep-dm)# exit
RouterA(config-pm-ep)# exit
RouterA(config-perf-meas)# delay-profile endpoint default
RouterA(config-pm-dm-ep)# probe
RouterA(config-pm-dm-ep-probe)# measurement-mode one-way
```

### Running Configuration

```
segment-routing
 traffic-eng
  segment-list SIDLIST1-Algo0
   index 10 mpls label 16005
   !
  segment-list SIDLIST2-FlexAlgo128
   index 10 mpls label 16085
   !
  !
 !
!
performance-measurement
 endpoint ipv4 1.1.1.5
  segment-list name SIDLIST1-Algo0
  !
  segment-list name SIDLIST2-FlexAlgo128
  !
  source-address ipv4 1.1.1.1
  delay-measurement
  !
 !
 delay-profile endpoint default
  probe
   measurement-mode one-way
  !
 !
!
```

### Verification

```
RouterA# show performance-measurement endpoint ipv4 1.1.1.5

--------------------------------------------------------------------------------
0/RSP0/CPU0
--------------------------------------------------------------------------------

Endpoint name: IPv4-1.1.1.5-vrf-default
  Source address              : 1.1.1.1
  VRF name                    : default
  Delay-measurement           : Enabled
  Description                 : Not set
  Profile Keys:
    Profile name              : default
    Profile type              : Endpoint Delay Measurement


  Segment-list                : None
  Delay Measurement session:
    Session ID          : 33554433
    Last advertisement:
      No advertisements have occured

    Next advertisement:
      Threshold check scheduled in 4 more probes (roughly every 120 seconds)
      No probes completed

    Current computation:
      Started at: Jul 19 2021 16:31:53.827 (15.844 seconds ago)
      Packets Sent: 6, received: 0
      Measured delays (uSec): avg: 0, min: 0, max: 0, variance: 0
      Next probe scheduled at: Jul 19 2021 16:32:22.957 (in 13.286 seconds)
      Next burst packet will be sent in 1.286 seconds
      Burst packet sent every 3.0 seconds

  Segment-list                : SIDLIST1-Algo0
  Delay Measurement session:
    Session ID          : 33554435
    Last advertisement:
      No advertisements have occured

    Next advertisement:
      Threshold check scheduled in 4 more probes (roughly every 120 seconds)
      No probes completed

    Current computation:
     Started at: Jul 19 2021 16:31:53.827 (15.844 seconds ago)
      Packets Sent: 4, received: 0
      Measured delays (uSec): avg: 0, min: 0, max: 0, variance: 0
      Next probe scheduled at: Jul 19 2021 16:32:22.957 (in 13.286 seconds)
      Next burst packet will be sent in 2.940 seconds
      Burst packet sent every 3.0 seconds


  Segment-list                : SIDLIST2-FlexAlgo128
  Delay Measurement session:
    Session ID          : 33554436
    Last advertisement:
      No advertisements have occured

    Next advertisement:
      Threshold check scheduled in 4 more probes (roughly every 120 seconds)
      No probes completed

    Current computation:
```

```
Started at: Jul 19 2021 16:31:53.827 (15.844 seconds ago)
 Packets Sent: 4, received: 0
 Measured delays (uSec): avg: 0, min: 0, max: 0, variance: 0
 Next probe scheduled at: Jul 19 2021 16:32:22.957 (in 13.286 seconds)
 Next burst packet will be sent in 2.940 seconds
 Burst packet sent every 3.0 seconds
```

### Use-Case 4: Liveness Detection Probe Toward an IP Endpoint

IP endpoint liveness detection leverages the loopback measurement-mode. The following workflow describes the sequence of events.

1. The sender creates and transmits the PM probe packets.

   The IP destination address (DA) on the probe packets is set to the loopback value of the sender itself.

   The transmit timestamp (T1) is added to the payload.

   The probe packet is encapsulated with the label corresponding to the endpoint.

2. The network delivers the PM probe packets following the LSP toward the endpoint.

3. The end-point receives the PM probe packets.

   Packets are forwarded back to the sender based on the forwarding entry associated with the IP DA of the PM probe packet. If an LSP exists, the probe packet is encapsulated with the label of the sender.

4. The sender node receives the PM probe packets.

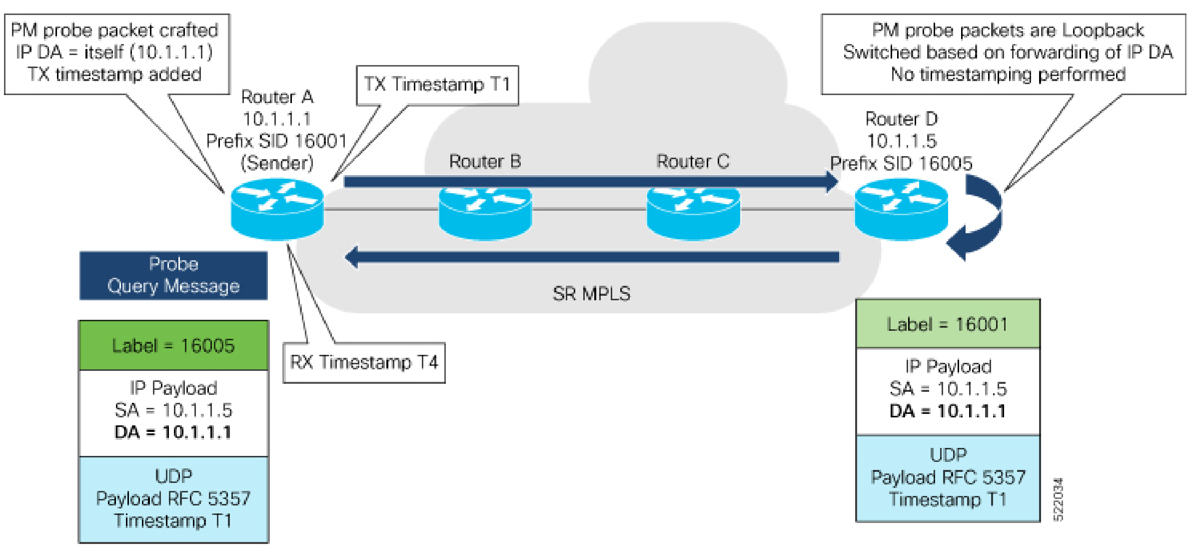
   The received timestamp (T4) stored.

   If the sender node doesn't receive the specified number of probe packets (based on the configured multiplier), the sender node declares the PM session as down.

The following figure illustrates a liveness detection probe toward an IP endpoint learned by the IGP. The network interconnecting the sender and reflector provides MPLS connectivity with Segment Routing.

The liveness detection multiplier is set to 5 to specify the number of consecutive missed probe packets before the PM session is declared as down.

*Figure 8: IP Endpoint Liveness Detection*



### Configuration

```
RouterA(config)# performance-measurement
RouterA(config-perf-meas)# endpoint ipv4 1.1.1.5
RouterA(config-pm-ep)# source-address ipv4 1.1.1.1
RouterA(config-pm-ep)# liveness-detection
RouterA(config-pm-ep-ld)# exit
RouterA(config-pm-ep)# exit
RouterA(config-perf-meas)# liveness-profile endpoint default
RouterA(config-pm-ld-ep)# liveness-detection
RouterA(config-pm-ld-ep-ld)# multiplier 5
RouterA(config-pm-ld-ep-ld)# exit
RouterA(config-pm-ld-ep)# probe
RouterA(config-pm-ld-ep-probe)# measurement-mode loopback
```

### Running Configuration

```
performance-measurement
 endpoint ipv4 1.1.1.5
  source-address ipv4 1.1.1.1
  liveness-detection
  !
 !
 liveness-profile endpoint default
  liveness-detection
   multiplier 5
  !
  probe
   measurement-mode loopback
  !
 !
!
end
```

### Verification

```
RouterA# show performance-measurement endpoint ipv4 1.1.1.5

--------------------------------------------------------------------------------
0/RSP0/CPU0
--------------------------------------------------------------------------------

Endpoint name: IPv4-1.1.1.5-vrf-default
  Source address              : 1.1.1.1
  VRF name                    : default
  Liveness Detection          : Enabled
  Profile Keys:
    Profile name              : default
    Profile type              : Endpoint Liveness Detection

  Segment-list                : None
  Session State: Down
  Missed count: 0
```

# SR Policy End-to-End Delay Measurement

| Feature Name | Release | Description |
|---|---|---|
| SR Policy End-to-End Delay Measurement | Release 7.5.2 | This feature allows you to monitor the end-to-end delay experienced by the traffic sent over an SR policy to ensure that the delay does not exceed the requested "upper-bound" and violate SLAs. |

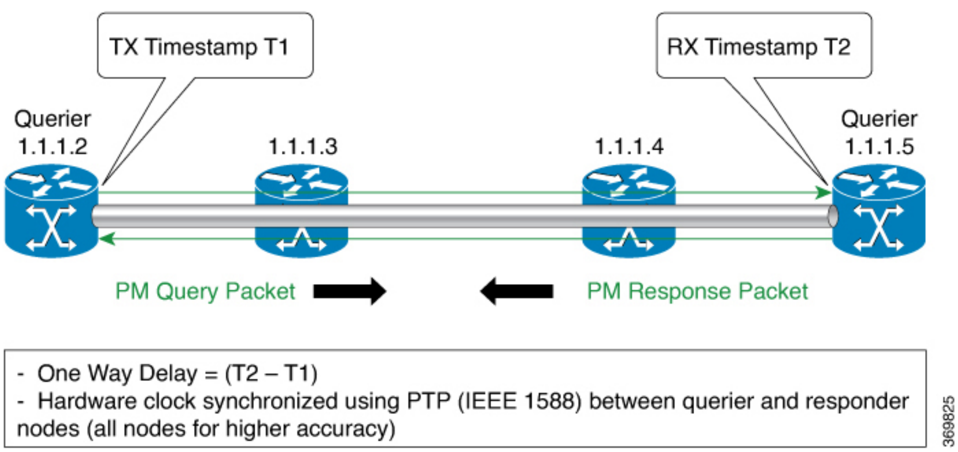
The PM for SR Policy uses the MPLS packet format defined in RFC 6374 or IP/UDP packet format defined in RFC 5357 (TWAMP-Light) for probes. The MPLS packet format requires the remote-side line card to be MPLS-capable.

The extended TE link delay metric (minimum-delay value) can be used to compute paths for SR policies as an optimization metric or as an accumulated delay bound.

There is a need to monitor the end-to-end delay experienced by the traffic sent over an SR policy to ensure that the delay does not exceed the requested "upper-bound" and violate SLAs. You can verify the end-to-end delay values before activating the candidate-path or the segment lists of the SR policy in forwarding table, or to deactivate the active candidate-path or the segment lists of the SR policy in forwarding table.

**Note** The end-to-end delay value of an SR policy will be different than the path computation result (for example, the sum of TE link delay metrics) due to several factors, such as queuing delay within the routers.

*Figure 9: Performance Measurement for SR Policy End-to-End Delay*




The PM query and response for end-to-end SR Policy delay can be described in the following steps:

1. The local-end router sends PM query packets periodically to the remote side once the egress line card on the router applies timestamps on packets.
2. The ingress line card on the remote-end router applies time-stamps on packets as soon as they are received.
3. The remote-end router sends the PM packets containing time-stamps back to the local-end router.

**4.** One-way delay is measured using the time-stamp values in the PM packet.

### Restrictions and Usage Guidelines for PM for SR Policy Delay

Hardware clocks must be synchronized between the querier and the responder nodes of the link using PTP for one-way delay measurement.

### Enable One-Way Delay Mode

This example shows how to enable one-way delay mode.

When one-way delay mode is enabled, an IP/UDP TLV (defined in RFC 7876) is added in the query packet to receive the PM reply via IP/UDP. Only two time-stamps (T1 and T2) defined in the RFC 6374 packets are used. Hardware clocks must be synchronized between querier and responder nodes (using PTP).

```
Router(config)# performance-measurement delay-profile sr-policy default
Router(config-pm-dm-intf)# probe measurement-mode one-way
Router(config-perf-meas)# exit
```

### Configuring Performance Measurement Parameters

This example shows how to configure performance-measurement parameters for SR policy delay as a global default profile. The default values for the different parameters in the PM for SR policy delay is given as follows:

- **probe**: The default mode for probe is one-way delay measurement.
- **burst-interval**: Interval for sending probe packet. The default value is 3000 milliseconds and the range is from 30 to 15000 milliseconds.
- **computation interval**: Interval for metric computation. Default is 30 seconds; range is 1 to 3600 seconds.
- **protocol**:
  - **twamp-light**: SR Policy delay measurement using RFC 5357 with IP/UDP encap. This is the default protocol.
  - **pm-mpls**: SR Policy delay measurement using RFC6374 with MPLS encap.
- **tos**: Type of Service
  - **dscp** *value*: The default value is 0 and the range is from 0 to 63.
  - **traffic-class** *value*: The default value is 0 and the range is from 0 to 7.
- **advertisement threshold-check**: The advertisement threshold-check has three types:
  - **average-delay**: Enable average-delay threshold-check.
  - **maximum-delay**: Enable maximum-delay threshold-check.
  - **minimum-delay**: Enable minimum-delay threshold-check.

**Note** The default value of periodic **advertisement threshold-check** is **maximum-delay**.

- **periodic advertisement**: Periodic advertisement is enabled by default.
- **periodic-advertisement interval**: The default value is 120 seconds and the interval range is from 30 to 3600 seconds.
- **periodic-advertisement threshold**: The default value of periodic advertisement threshold is 10 percent and the range is from 0 to 100 percent.
- **periodic-advertisement minimum-change**: The default value is 500 microseconds (usec) and the range is from 0 to 100000 microseconds.
- **accelerated advertisement**: Accelerated advertisement is disabled by default.
- **accelerated-advertisement threshold**: The default value is 20 percent and the range is from 0 to 100 percent.
- **accelerated-advertisement minimum**: The default value is 500 microseconds and the range is from 1 to 100000 microseconds.

```
Router(config)# performance-measurement delay-profile sr-policy default
Router(config-pm-dm-srpolicy)# probe
Router(config-pm-dm-srpolicy-probe)# tx-interval 60000
Router(config-pm-dm-srpolicy-probe)# computation-interval 60
Router(config-pm-dm-srpolicy-probe)# protocol twamp-light
Router(config-pm-dm-srpolicy-probe)# tos dscp
Router(config-pm-dm-srpolicy-probe)# exit

Router(config-pm-dm-srpolicy)# advertisement
Router(config-pm-dm-srpolicy-adv)# periodic
Router(config-pm-dm-srpolicy-adv-per)# interval 60
Router(config-pm-dm-srpolicy-adv-per)# minimum-change 1000
Router(config-pm-dm-srpolicy-adv-per)# threshold 20
Router(config-pm-dm-srpolicy-adv-per)# exit

Router(config-pm-dm-srpolicy-adv)# accelerated
Router(config-pm-dm-srpolicy-adv-acc)# minimum-change 1000
Router(config-pm-dm-srpolicy-adv-acc)# threshold 10
Router(config-pm-dm-srpolicy-adv-acc)# exit

Router(config-pm-dm-srpolicy-adv)# threshold-check minimum-delay
Router(config-pm-dm-srpolicy-adv)# exit
Router(config-pm-dm-srpolicy)#
```

### Configure the UDP Destination Port

Configuring the UDP port for TWAMP-Light protocol is optional. By default, PM uses port 862 as the TWAMP-reserved UDP destination port.

The UDP port is configured for each PM measurement probe type (delay, loss, protocol, authentication mode, etc.) on querier and responder nodes. If you configure a different UDP port, the UDP port for each PM measurement probe type must match on the querier and the responder nodes.

**Note** The same UDP destination port is used for delay measurement for links and SR Policy.

This example shows how to configure the UDP destination port.

```
Router(config)# performance-measurement
Router(config-perf-meas)# protocol twamp-light
Router(config-pm-protocol)# measurement delay unauthenticated
Router(config-pm-proto-mode)# querier-dst-port 12000
```

### Enable Performance Measurement for SR Policy

This example shows how to enable PM for SR policy delay for a specific policy.

```
Router(config)# segment-routing traffic-eng
Router(config-sr-te)# policy foo
Router(config-sr-te-policy)# performance-measurement
Router(config-sr-te-policy-perf-meas)# delay-measurement
```

### SR Policy Probe IP/UDP ECMP Hashing Configuration

This example shows how to configure SR Policy ECMP IP-hashing mode.

- The destination IPv4 address 127.x.x.x – 127.y.y.y is used in the Probe messages to take advantages of 3-tuple IP hashing (source-address, destination-address, and local router ID) for ECMP paths of SR-MPLS Policy.

**Note** The destination IPv4 address must be 127/8 range (loopback), otherwise it will be rejected.

- One PM session is always created for the actual endpoint address of the SR Policy.
- You can specify the number of IP addresses to sweep. The range is from 0 (default, no sweeping) to 128.
- Platforms may have a limitation for large label stack size to not check IP address for hashing.

```
Router(config)# performance-measurement delay-profile sr-policy default
Router(config-pm-dm-srpolicy)# probe
Router(config-pm-dm-srpolicy-probe)# sweep
Router(config-pm-dm-srpolicy-probe-sweep)# destination ipv4 127.0.0.1 range 28
```

# SR Policy Liveness Monitoring

*Table 6: Feature History Table*

| Feature Name | Release Information | Feature Description |
| --- | --- | --- |
| SR Policy Liveness Monitoring | Release 7.5.2 | This feature allows you to verify end-to-end traffic forwarding over an SR Policy candidate path by periodically sending performance monitoring packets. |

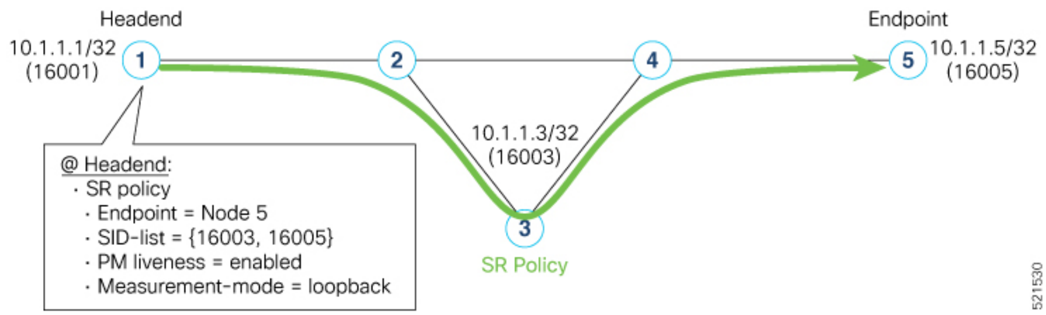
SR Policy liveness monitoring allows you to verify end-to-end traffic forwarding over an SR Policy candidate path by periodically sending probe messages. The head-end router sends PM packets to the SR policy's endpoint router, which sends them back to the head-end without any control-plane dependency on the endpoint router.

The following are benefits to using SR-PM liveness monitoring:

- Allows both liveness monitoring and delay measurement using a single-set of PM packets as opposed to running separate monitoring sessions for each purpose. This improves the overall scale by reducing the number of PM sessions required.
- Eliminates network and device complexity by reducing the number of monitoring protocols on the network (for example, no need for Bidirectional Failure Detection [BFD]). It also simplifies the network and device operations by not requiring any signaling to bootstrap the performance monitoring session.
- Improves interoperability with third-party nodes because signaling protocols aren't required. In addition, it leverages the commonly supported TWAMP protocol for packet encoding.
- Improves liveness detection time because PM packets aren't punted on remote nodes
- Provides a common solution that applies to data-planes besides MPLS, including IPv4, IPv6, and SRv6.

The workflow associated with liveness detection over SR policy is described in the following sequence.

Consider an SR policy programmed at head-end node router 1 towards end-point node router 5. This SR policy is enabled for liveness detection using the loopback measurement-mode.




- **A**: The head-end node creates and transmits the PM probe packets.

  The IP destination address (DA) on the probe packets is set to the loopback value of the head-end node itself.

  A transmit (Tx) timestamp is added to the payload.

  Optionally, the head-end node may also insert extra encapsulation (labels) to enforce the reverse path at the endpoint node.

  Finally, the packet is injected into the data-plane using the same encapsulation (label stack) of that of the SR policy being monitored.

- **B**: The network delivers the PM probe packets as it would user packet for the SR policy.
- **C**: The end-point node receives the PM probe packets.

  Packets are switched back based on the forwarding entry associated with the IP DA of the packet. This would typically translate to the end-point node pushing the prefix SID label associated with the head-end node.

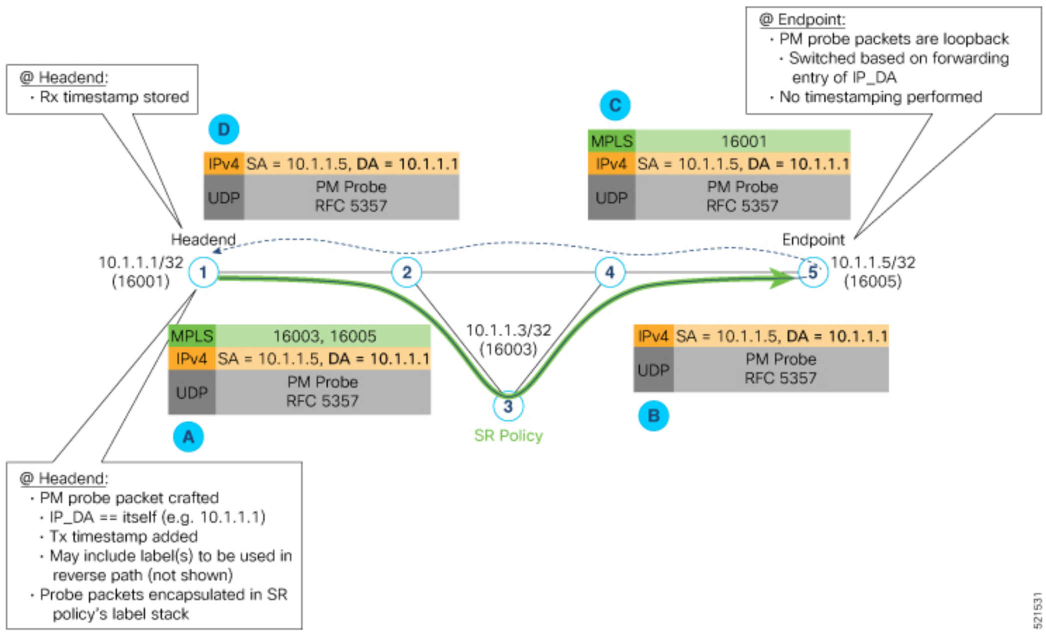
If the head-end node inserted label(s) for the reverse path, then the packets are switched back at the end-point node based on the forwarding entry associated with the top-most reverse path label.

- **D**: Headend node receives the PM probe packets.

  A received (Rx) timestamp stored.

  If the head-end node receives the PM probe packets, the head-end node assume that the SR policy active candidate path is up and working.

  If the head-end node doesn't receive the specified number of consecutive probe packets (based on configured multiplier), the head-end node assumes the candidate path is down and a configured action is trigerred.




### Usage Guidelines and Limitations

The following usage guidelines and limitations apply:

- Liveness-detection and delay-measurement aren't supported together
- When liveness-profile isn't configured, SR Policies use the default values for the liveness-detection profile parameters.

### Configuring SR Policy Liveness Monitoring

Configuring SR Policy liveness monitoring involves the following steps:

- Configuring a performance measurement liveness profile to customize generic probe parameters
- Enabling liveness monitoring under SR Policy by associating a liveness profile, and customizing SR policy-specific probe parameters

### Configuring Performance Measurement Liveness Profile

Liveness monitoring parameters are configured under sub-mode. The following parameters are configurable:

- **liveness-profile sr-policy {default | name** *name*}
- **probe**: Configure the probe parameters.
- **measurement-mode**: Liveness detection must use loopback mode (see Measurement Mode topic within this guide).
- **burst interval**: Interval for sending probe packet. The default value is 3000 milliseconds and the range is from 30 to 15000 milliseconds.
- **tos dscp** *value*: The default value is 48 and the range is from 0 to 63. You can modify the DSCP value of the probe packets, and use this value to priortize the probe packets from headend to tailend.
- **sweep destination ipv4 127.***x.x.x* **range** *range*: Configure SR Policy ECMP IP-hashing mode. Specifiy the number of IP addresses to sweep. The range is from 0 (default, no sweeping) to 128. The option is applicable to IPv4 packets.

**Note** The destination IPv4 headendaddress 127.x.x.x – 127.y.y.y is used in the Probe messages to take advantages of 3-tuple IP hashing (source-address, destination-address, and local router ID) for ECMP paths of SR-MPLS Policy.

The destination IPv4 address must be 127/8 range (loopback), otherwise it will be rejected.

**Note** One PM session is always created for the actual endpoint address of the SR Policy.

- **liveness-detection**: Configure the liveness-detection parameters:
- **multiplier**: Number of consecutive missed probe packets before the PM session is declared as down. The range is from 2 to 10, and the default is 3.

**Note** The detection-interval is equal to (burst-interval * multiplier).

### Enabling Liveness Monitoring under SR Policy

Enable liveness monitoring under SR Policy, associate a liveness-profile, and configure SR Policy-specific probe parameters under the **segment-routing traffic-eng policy performance-measurement** sub-mode. The following parameters are configurable:

- **liveness-detection**: Enables end-to-end SR Policy Liveness Detection for all segment-lists of the active and standby candidate-path that are in the forwarding table.
- **liveness-profile name** *name*: Specifies the profile name for named profiles.
- **invalidation-action {down | none}**:

    - **Down (default)**: When the PM liveness session goes down, the candidate path is immediately operationally brought down.
    - **None**: When the PM liveness session goes down, no action is taken. If logging is enabled, the failure is logged but the SR Policy operational state is not modified.
- **logging session-state-change**: Enables Syslog messages when the session state changes.
- **reverse-path label** {*BSID-value* | *NODE-SID-value* | *ADJACENCY-SID-value*}: Specifies the MPLS label to be used for the reverse path for the reply. If you configured liveness detection with ECMP hashing, you must specify the reverse path. The default reverse path uses IP Reply.
    - *BSID-value*: The Binding SID (BSID) label for the reverse SR Policy. (This is practical for manual SR policies with a manual BSID.)
    - *NODE-SID-value*: The Node SID is a segment type that represents the ECMP-aware shortest path to reach a particular IP prefix from any IGP topology location
    - *ADJACENCY-SID-value*: The absolute SID label of the (local) Sender Node to be used for the reverse path for the reply.

### Configuration Examples

#### Configure a Default SR-Policy PM Liveness-Profile

The following example shows a default sr-policy liveness-profile:

```
RP/0/RSP0/CPU0:ios(config)# performance-measurement
RP/0/RSP0/CPU0:ios(config-perf-meas)# liveness-profile sr-policy default
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy)# probe
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# measurement-mode loopback
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# burst-interval 1500
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# tos dscp 52
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# exit
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy)# liveness-detection
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-ld)# multiplier 5
```

#### Running Configuration:

```
performance-measurement
 liveness-profile sr-policy default
  liveness-detection
   multiplier 5
  !
  probe
   tos dscp 52
   measurement-mode loopback
   burst-interval 1500
  !
 !
!
end
```

#### Configure a Named (Non-Default) SR-Policy PM Liveness-Profile

The following example shows a named sr-policy liveness-profile:

```
RP/0/RSP0/CPU0:ios(config)# performance-measurement
RP/0/RSP0/CPU0:ios(config-perf-meas)# liveness-profile sr-policy name sample-profile
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy)# probe
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# measurement-mode loopback
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# burst-interval 1500
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# tos dscp 52
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# exit
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy)# liveness-detection
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-ld)# multiplier 5
```

**Running Configuration:**

```
performance-measurement
 liveness-profile sr-policy name sample-profile
  liveness-detection
   multiplier 5
  !
  probe
   tos dscp 52
   measurement-mode loopback
   burst-interval 1500
  !
 !
!
end
```

**Configure a SR-Policy PM Liveness-Profile with Sweep Parameters**

The following example shows a named sr-policy liveness-profile with sweep parameters:

```
RP/0/RSP0/CPU0:ios(config)# performance-measurement
RP/0/RSP0/CPU0:ios(config-perf-meas)# liveness-profile sr-policy name sample-profile
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy)# probe
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# measurement-mode loopback
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# burst-interval 1500
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# tos dscp 52
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# sweep
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe-sweep)# destination ipv4 127.0.0.1 range 25
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe-sweep)# exit
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# exit
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy)# liveness-detection
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-ld)# multiplier 5
```

**Running Configuration**

```
performance-measurement
 liveness-profile sr-policy name sample-profile
  liveness-detection
   multiplier 5
  !
  probe
   tos dscp 52
   sweep
    destination ipv4 127.0.0.1 range 25
   !
   measurement-mode loopback
   burst-interval 1500
  !
 !
!
end
```

**Enable Liveness Monitoring Under SR Policy**

The following example shows how to enable liveness monitoring under SR Policy, associate a liveness-profile, and configure the invalidation action:

```
RP/0/RSP0/CPU0:ios(config)# segment-routing traffic-eng
RP/0/RSP0/CPU0:ios(config-sr-te)# policy FOO
RP/0/RSP0/CPU0:ios(config-sr-te-policy)# performance-measurement
RP/0/RSP0/CPU0:ios(config-sr-te-policy-perf-meas)# liveness-detection
RP/0/RSP0/CPU0:ios(config-sr-te-policy-live-detect)# liveness-profile name sample-profile
RP/0/RSP0/CPU0:ios(config-sr-te-policy-live-detect)# invalidation-action none
```

### Running Config

```
segment-routing
 traffic-eng
  policy FOO
   performance-measurement
    liveness-detection
     liveness-profile name sample-profile
     invalidation-action none
    !
   !
  !
 !
!
end
```

### Enable Liveness Monitoring under SR Policy with Optional Parameters

The following example shows how to enable liveness monitoring under SR Policy, associate a liveness-profile, and configure reverse path label and session logging:

```
RP/0/RSP0/CPU0:ios(config)# segment-routing traffic-eng
RP/0/RSP0/CPU0:ios(config-sr-te)# policy BAA
RP/0/RSP0/CPU0:ios(config-sr-te-policy)# performance-measurement
RP/0/RSP0/CPU0:ios(config-sr-te-policy-perf-meas)# liveness-detection
RP/0/RSP0/CPU0:ios(config-sr-te-policy-live-detect)# liveness-profile name sample-profile
RP/0/RSP0/CPU0:ios(config-sr-te-policy-live-detect)# invalidation-action down
RP/0/RSP0/CPU0:ios(config-sr-te-policy-live-detect)# logging session-state-change
RP/0/RSP0/CPU0:ios(config-sr-te-policy-live-detect)# exit
RP/0/RSP0/CPU0:ios(config-sr-te-policy-perf-meas)# reverse-path label 16001
```

### Running Config

```
segment-routing
 traffic-eng
  policy BAA
   performance-measurement
    liveness-detection
     logging
      session-state-change
     !
     liveness-profile name sample-profile
     invalidation-action down
    !
    reverse-path
     label 16001
    !
   !
  !
 !
!
end
```