



Configure Performance Measurement

Network performance metrics is a critical measure for traffic engineering (TE) in service provider networks. Network performance metrics include the following:

- Packet loss
- Delay
- Delay variation
- Bandwidth utilization

These network performance metrics provide network operators information about the performance characteristics of their networks for performance evaluation and help 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 Segment Routing Performance Measurement (SR-PM) feature to monitor the network metrics for links and 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 default profiles for different types of delay measurements. Use the "interfaces" delay profile type for link-delay measurement. The "sr-policy" delay profile type is used for SR policy delay measurements. Delay profile allows you to schedule probe and configure metric advertisement parameters for delay measurement.
Protocols	Two-Way Active Measurement Protocol (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, session counters, and packet advertisement counters.

The following are the means by which you can measure the performance of your network:

- [Liveness Monitoring, on page 2](#)
- [Delay Measurement, on page 15](#)

Liveness Monitoring

Table 2: Feature History Table

Liveness refers to the ability of the network to confirm that a specific path, segment, or a node is operational and capable of forwarding packets. Liveness checks are essential for maintaining network availability and reliability.

Benefits

- **Fault Detection:** You can quickly identify if a device is down, which allows for immediate response and troubleshooting.
- **Load Balancing:** You can identify if the devices in a network are live, so work can be distributed more evenly across the network, preventing overloading of specific components and improving overall performance.
- **System Health:** You can provide an ongoing snapshot of a system's health, helping to identify potential issues before they become significant problems.
- **Maintenance Planning:** Liveness information can also help with maintenance planning, as system administrators can understand which components are live or down and plan maintenance and downtime accordingly without significant disruption to services.
- **Security:** Regular liveness checks can also play a role in maintaining network security. Administrators can take proactive steps to mitigate the damage and prevent future incidents by identifying unusual activity that might indicate a security breach or attack.

You can determine liveness for SR Policy and IP Endpoint.

IP Endpoint Liveness Monitoring

Table 3: 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). This feature is supported on IPv4, IPv6, and MPLS data planes.

The Segment Routing Performance Measurement (SR-PM) for IP endpoint liveness is a type of node liveness that involves testing whether an IP endpoint or a device identified by an IP address is available to send and receive data.

IP endpoint liveness is verified by sending a request to the IP address of the endpoint and waiting for a response. The probe could be an ICMP echo request (Ping), a TCP packet, a UDP packet, or any other type of packet that the endpoint would respond to.

- If a response is received, the endpoint is considered *live*.
- If no response is received within a certain time frame, the endpoint is considered *down* or *unreachable*.

IP endpoint dynamically measures the liveness towards a specified IP endpoint. IP endpoints can be located in a default or nondefault VRFs. IP endpoint is any device in the network a device identified by an IP address.

Liveness of an IP endpoint is verified by sending a request to the IP address of the endpoint and waiting for a response, which is referred to as a probe.

The endpoint of a probe is defined by an IP address, which can be either IPv4 or IPv6. This IP address can be any address that the sender can reach, such as a local interface or a remote node or host, either within an operator's network or accessible via a VRF.

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 IP address of the endpoint 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.
- 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, the sender node uses the configured VRF for the endpoint address. In the return path, the reflector node derives the VRF based on which incoming VRF label the probe packet is received with.

You can configure the following parameters in the **performance-measurement** command:

- **Endpoint:** The endpoint of a probe is defined by an IP address, which can be either IPv4 or IPv6. This IP address can be any address that the sender can reach, such as a local interface or a remote node or host, either within an operator's network or accessible via a VRF.

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.

Use the **performance-measurement endpoint** command to configure a probe endpoint source and destination addresses on a sender node.

- **VRF:** You can define the endpoint point IP address belonging to a specific VRF. Use the **performance-measurement endpoint {ipv4 | ipv6} ip_addr [vrf WORD]** command to configure an endpoint to define the VRF. Endpoint segment list configuration is not supported under nondefault VRF.
 - 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

- **Source address:** You can define the source of the endpoint using the endpoint specific source address and the global source address.

Global source address configuration is applied to all the endpoints when the endpoint specific source address configuration isn't specified. endpoint specific configuration overrides all the global source address configuration for those specific endpoints for which source addresses are configured.

For Micro-SID configuration for IPv4 endpoint sessions, if IPv6 global source address is configured, then it applies the configured global IPv6 source address for the IPv6 header in the SRv6 packet. If IPv6 global address is not configured, then It does not form a valid SRv6 packet.

You can use the **source-address** keyword under the **performance-measurement** command to define the global source address or use the keyword under **performance-measurement endpoint** to define endpoint specific source address.

Usage Guidelines and Limitations

- Liveness session without segment list for an endpoint in a non-default VRF is not supported.
- SR Performance Measurement endpoint session over BVI interface is not supported.

IP Endpoint Liveness Detection in an SR MPLS Network

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.

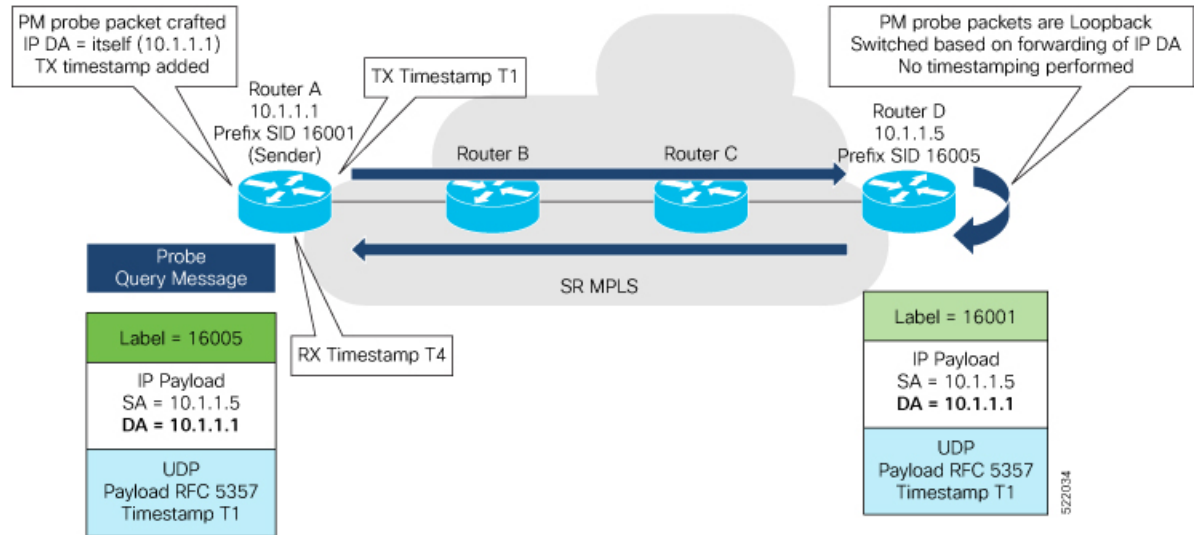
The received timestamp (T4) stored.

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 1: IP Endpoint Liveness Detection



Configuration Example

```
RouterA(config)# performance-measurement
RouterA(config-perf-meas)# endpoint ipv4 10.1.1.5
RouterA(config-pm-ep)# source-address ipv4 10.1.1.1
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
```

Running Configuration

```
performance-measurement
 endpoint ipv4 10.1.1.5
   source-address ipv4 10.1.1.1
   liveness-detection
   !
   !
 liveness-profile endpoint default
   liveness-detection
     multiplier 5
   !
 !
 !
end
```

Verification

```
RouterA# show performance-measurement endpoint ipv4 10.1.1.5
```

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

```

Endpoint name: IPv4-10.1.1.5-vrf-default
Source address      : 10.1.1.1
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 Liveness Monitoring

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
SR Performance Measurement Named Profiles	Release 7.3.1	<p>You can use this feature to create specific performance measurement delay and liveness profiles, and associate it with an SR policy.</p> <p>This way, a delay or liveness profile can be associated with a policy for which the performance measurement probes are enabled, and performance measurement is precise, and enhanced.</p> <p>The performance-measurement delay-profile sr-policy command was updated with the name <i>profile</i> keyword-argument combination.</p> <p>The performance-measurement liveness-profile sr-policy command was updated with the name <i>profile</i> keyword-argument combination.</p> <p>The performance-measurement delay-measurement command was updated with delay-profile name <i>profile</i>.</p> <p>The performance-measurement liveness-detection command was updated with liveness-profile name <i>profile</i>.</p>

Feature Name	Release Information	Feature Description
SR Policy Liveness Monitoring	Release 7.3.1	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 performance monitoring (PM) packets. 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.

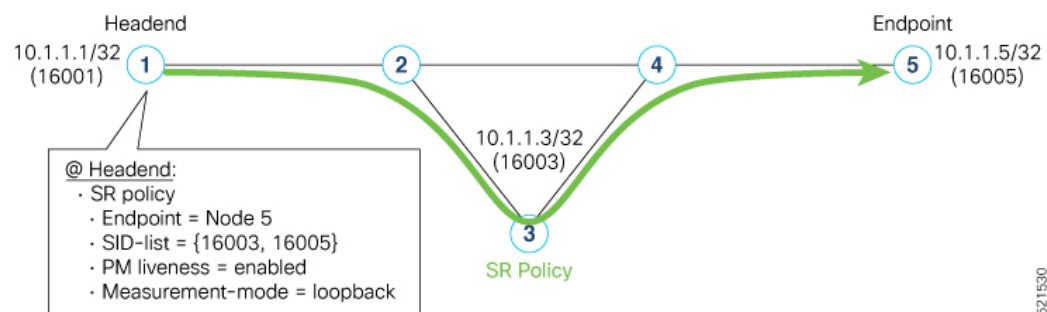
The following are benefits to using SR-PM liveness monitoring:

- 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.

How it works?

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 traffic over 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.

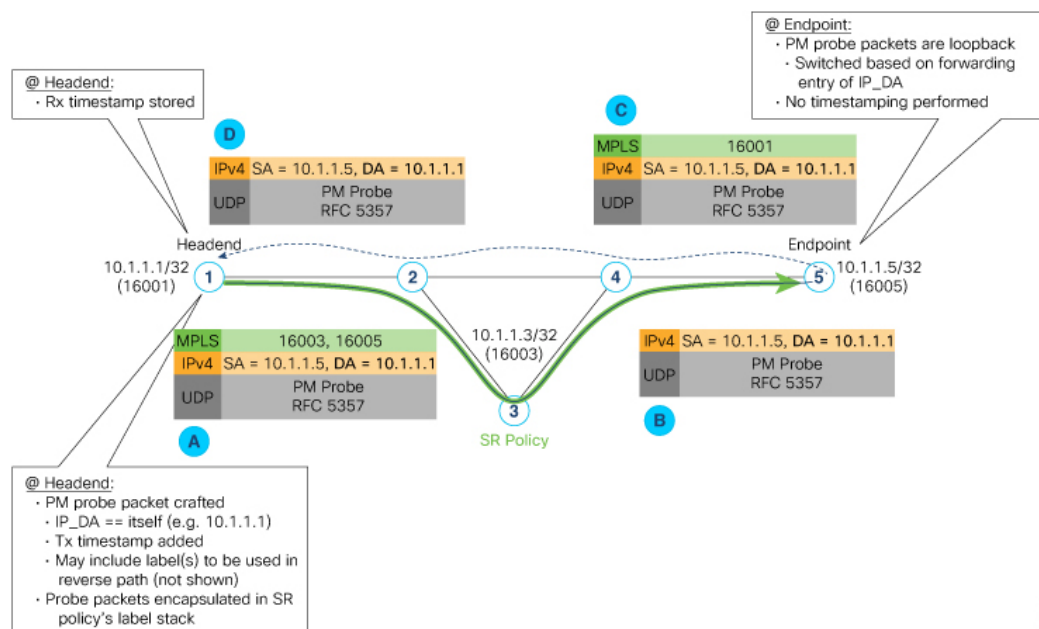
If the head-end node inserted label(s) for the reverse path, then the packets are switched back at the end-point node based on the forwarding entry associated with the top-most reverse path label.

- **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 triggered.



Usage Guidelines and Limitations

The following usage guidelines and limitations apply:

- SR-PM liveness-detection over SR Policy is supported on manually configured SR Policies and On-Demand SR Policies (ODN).
- SR-PM liveness-detection over SR Policy is not supported on PCE-initiated SR Policies.

- SR-PM 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.

Configure SR Policy Liveness Monitoring in an MPLS Network

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

Liveness monitoring parameters are configured under **performance-measurement liveness-profile** sub-mode. The following parameters are configurable:

- **liveness-profile** {**sr-policy default** | **name** *name*}

Parameters defined under the **sr-policy default** liveness-profile apply to any SR policy with liveness monitoring enabled and that does not reference a non-default (named) liveness-profile.

- **probe**: Configure the probe parameters.
- **tx-interval**: Interval for sending probe packet. The default value is 3000000 microseconds and the range is from 30000 to 15000000 microseconds.
- **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 prioritize the probe packets from headend to tailend.
- **sweep destination ipv4 127.x.x.x range range**: Configure SR Policy ECMP IP-hashing mode. Specify 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 headend 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.

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 (tx-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 isn't modified.
- **logging session-state-change**: Enables Syslog messages when the session state changes.
- **reverse-path label {BSID-value | NODE-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 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)# tx-interval 150000
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
  tx-interval 150000
 !
```

```

!
!
end

```

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

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

```

Router(config)# performance-measurement
Router(config-perf-meas)# liveness-profile name sample-profile
Router(config-pm-ld-profile)# probe
Router(config-pm-ld-probe)# tx-interval 150000
Router(config-pm-ld-probe)# tos dscp 52
Router(config-pm-ld-probe)# exit

Router(config-pm-ld-profile)# liveness-detection
Router(config-pm-ld-profile-ld)# multiplier 5
Router(config-pm-ld-profile-ld)#commit

```

Running Configuration:

```

performance-measurement
  liveness-profile name sample-profile
  liveness-detection
    multiplier 5
  !
  probe
    tos dscp 52
    tx-interval 150000
  !
!
end

```

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

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

```

Router(config)# performance-measurement
Router(config-perf-meas)# liveness-profile name sample-profile
Router(config-pm-ld-profile)# probe
Router(config-pm-ld-probe)# tx-interval 150000
Router(config-pm-ld-probe)# tos dscp 52
Router(config-pm-ld-probe)# sweep
Router(config-pm-ld-probe-sweep)# destination ipv4 127.0.0.1 range 25
Router(config-pm-ld-probe-sweep)# exit
Router(config-pm-ld-probe)# exit

Router(config-pm-ld-profile)# liveness-detection
Router(config-pm-ld-profile-ld)# multiplier 5
Router(config-pm-ld-profile-ld)#commit

```

Running Configuration

```

performance-measurement
  liveness-profile name sample-profile
  liveness-detection
    multiplier 5
  !
  probe
    tos dscp 52
    sweep
      destination ipv4 127.0.0.1 range 25
    !
    tx-interval 150000
  !
!
end

```

```

!
!
!
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
```

SR Policy Liveness Monitoring - Hardware Offloading

Table 5: Feature History Table

Feature Name	Release	Description
SR Policy Liveness Monitoring - Hardware Offloading	Release 7.10.1	<p>You can now hardware offload the liveness monitoring in performance measurement to the router hardware, which is the Network Processing Unit (NPU). This feature helps you optimize and scale the measurement operation, helping you meet delay-bound Service Level Agreements (SLAs). Previously, this feature was software driven.</p> <p>The feature introduces a new keyword npu-offload under the performance-measurement liveness-profile name <i>liveness profile</i> command.</p> <p>Note This feature is not supported on NC57 series of LC cards and routers.</p>

Performance Measurement (PM) hardware offload feature allows the offload of PM liveness monitoring session to the Network Processing Unit (NPU) on the platform, which considerably improves scale and reduces the overall network convergence detection time.

This improvement is done by sending rapid failure detection probes (messages) and detecting policy or path failures quickly and help routing protocols in recalculating the routing table.

This feature is required in order to quickly react on delay-bound Service Level Agreement (SLAs), for example 5G low-latency, where SRTE policy can quickly re-optimize once the SLA is violated.

Advantages of the PM Hardware Offloading feature are as listed:

- Probes are sent every 3.3 milliseconds
- Complete liveness of the endpoint is now reduced to 10ms from 50ms when the operator configures the multiplier to be 3 (10ms = 3.3ms * 3).
- Currently, the hardware offload supports only liveness monitoring .



Note The hardware offload does not support delay and loss measurement yet.

Usage Guidelines and Limitations

The following usage guidelines and limitations apply:

- The NPU offload generates PM probes (Packets Per Second) with a maximum limit. PPS is directly proportional with the number of sessions and transmit interval. If the PPS exceeds the limit supported

by the offload engine, the stretch algorithm activates. This algorithm doubles the transmit interval until the PPS is within the supported limit. Use 'show performance-measurement pps' command to verify the maximum probes per second (PPS) supported by offload engine and total pps currently in use.

Configuration Example



Note The **Hw-module profile offload 4** command is a prerequisite for LC CPU sessions to work.

Once you use the 'hw-module profile offload 4' command, the Bidirectional Forwarding Detection for IPv6 (BFDv6) will not work even if the Performance Measurement sessions are hosted only on LC CPU and not offloaded to the offload processor.

Use the 4th option **4 PM-HW-Offload and Bsync** in **hw-module profile offload** command to configure the Hardware Profile Offloading.

```
Router(config)#hw-module profile offload ?
 1 BFDv6 and Bsync
 2 BFDv6 and Route download
 3 Route download and Bsync
 4 PM-HW-Offload and Bsync
```

Reload the router to apply the hardware offload profile. After reloading, the Performance Measurement application loads on the offload engine.

The following example allows you to enable Performance Measurement Liveness Hardware (NPU) offload in the SR environment.

```
Router(config)#performance-measurement
Router(config-perf-meas)#liveness-profile name hwo_profile
Router(config-pm-ld-profile)#npu-offload
Router(config-pm-ld-profile-npu-offload)#enable
Router(config-pm-ld-profile-npu-offload)#commit
```

Running Configuration

The running configuration for this feature is as shown:

```
performance-measurement
  liveness-profile name
    npu-offload
    enable
  !
  !
  !
```

Verification

Use the show command to verify the running configuration as shown:

```
Router# show performance-measurement sessions detail

Transport type : SR Policy
Measurement type : Liveness Detection
Policy name : srte_c_90005_ep_10.2.2.2
```

```
Color : 90005
Endpoint : 10.2.2.2
Instance : 7
preference : 20
Protocol-origin : Configured
Discriminator : 20
Segment-list : route_12_2
Atomic path:
Hops : 10.2.2.2
Session ID : 45
Trace ID : 3111803555
NPU Offloaded session : True
NPU number : 0
NPU session state : Session created
Retry count : 0
Last NPU notification:
Session state : Up
Timestamp : Feb 28 2023 16:28:09.411
Timestamping Enabled : True
Liveness Detection: Enabled
Session State: Up
Last State Change Timestamp: Feb 28 2023 16:28:09.411
Missed count: 0
```

Delay Measurement

Delay measurement is a mechanism used to measure the latency or delay experienced by data packets when they traverse a network.

The PM for delay measurement 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.

Benefits

- **Network Troubleshooting:** You can quickly and easily identify areas in your network with high delay and resolve network problems using delay measurement.
- **Network Planning and Optimization:** You can easily understand the performance of your network under various conditions and design a network that can handle expected traffic loads.
- **Quality of Service (QoS):** You can ensure quality of service standards are being met by continuously monitoring the delay in your network.

Supported Delay Measurement Methods

You can measure delay using the following methods:

- [Link Delay Measurement, on page 18](#) Use to monitor delay experienced by data packets in a single link or path between two nodes in a network.

- **Delay Measurement for IP Endpoint:** Use to monitor the amount of time it takes for a data packet to travel from a source device to a specific IP endpoint within a network.
- **SR Policy End-to-End Delay Measurement** , on page 45: Use to to monitor the end-to-end delay experienced by the traffic sent over an SR policy.

Measurement Modes

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

Table 6: Measurement Mode Requirements

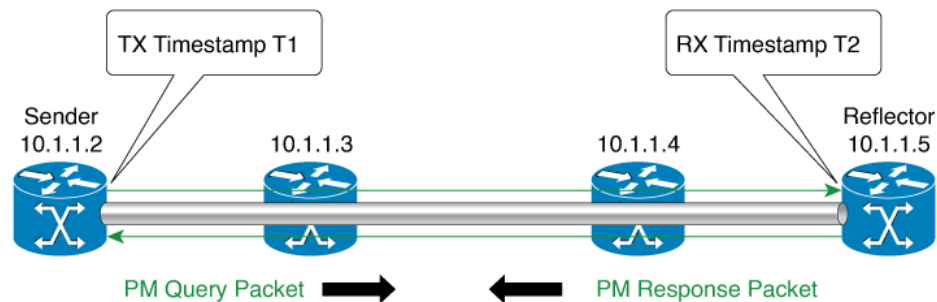
Measurement Mode	Sender: PTP-Capable HW and HW Timestamping	Reflector: PTP-Capable HW and HW Timestamping	PTP Clock Synchronization between Sender and Reflector
One-way	Required	Required	Required
Two-way	Required	Required	Not Required
Loopback	Required	Not Required	Not Required

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 2: One-Way



- One Way Delay = $(T2 - T1)$
- Hardware clock synchronized using PTP (IEEE 1588) between sender and reflector nodes (all nodes for higher accuracy)

521501

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.

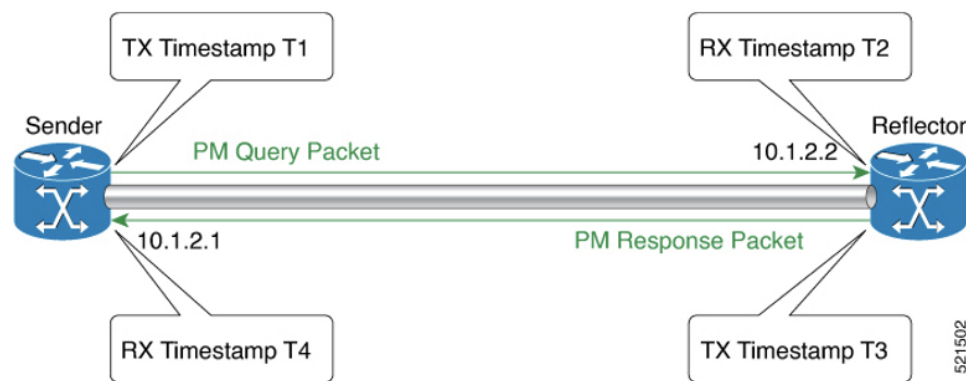
2. The ingress line card on the remote-end router applies time-stamps on packets as soon as they are received.
3. The remote-end router sends the PM packets containing time-stamps back to the local-end router.
4. One-way delay is measured using the time-stamp values in the PM packet.

Two-Way Measurement Mode

Two-way measurement 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 3: 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.

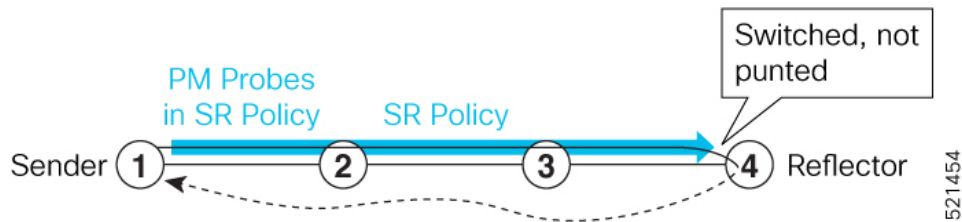
Loopback Measurement Mode

Loopback 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.

Delay measurements in Loopback mode are calculated as follows:

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

Figure 4: 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 = $T4 - T1$.

Link Delay Measurement



Note From Cisco IOS XR Release 7.6.1 onwards, Cisco NCS 540 routers support the following features:

- Link Delay Measurement
- Named Profiles
- Static Delay Value on an Interface

Table 7: Feature History Table

Feature Name	Release Information	Feature Description
Link Delay Measurement with IPv6 Link Local Address	Release 7.3.1	The performance measurement for link delay determines the source and destination IP addresses used in the OAM packet based on the IP address of the interface, where the delay measurement operation is enabled. This feature enables using the IPv6 link-local address as the OAM packet source IP address, when no IPv4 or IPv6 address is configured in the interface.

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.

Usage Guidelines and Restrictions for PM for Link Delay

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

- 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). See [Measurement Modes, on page 16](#) for more information.
- **protocol:** Interface delay measurement using RFC 5357 with IP/UDP encap (TWAMP-Light).
- **tx-interval:** Interval for sending probe packet. The default value is 3000000 microseconds and the range is from 30000 to 15000000 microseconds.
- **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:** Checks the minimum-delay metric change for threshold crossing for periodic advertisement. The default value is 10 percent and the range is from 0 to 100 percent.
- **periodic-advertisement minimum change:** The default value is 1000 microseconds (usec) and the range is from 0 to 100000 microseconds.
- **accelerated advertisement:** Accelerated advertisement is disabled by default.
- **accelerated-advertisement threshold:** Checks the minimum-delay metric change for threshold crossing for accelerated advertisement. The default value is 20 percent and the range is from 0 to 100 percent.
- **accelerated-advertisement minimum change:** The default value is 500 microseconds and the range is from 0 to 100000 microseconds.

```

RP/0/0/CPU0:router(config)# performance-measurement delay-profile interfaces default
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)# tx-interval 30000
RP/0/0/CPU0:router(config-pm-dm-intf-probe)# computation-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 for delay.

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 for delay.

```

Router(config)# performance-measurement
Router(config-perf-meas)# protocol twamp-light
Router(config-pm-protocol)# measurement delay unauthenticated
Router(config-pm-protocol-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)# next-hop ipv4 10.10.10.2 // Optional IPv4 or IPv6
next-hop address
RP/0/0/CPU0:router(config-pm-intf)# delay-measurement
RP/0/0/CPU0:router(config-pm-intf-dm)# exit

```

The source and destination IP addresses used in the OAM packet are determined by the IP address present on the interface where the delay-measurement operation is enabled and the setting of the optional **next-hop** address.

When the **next-hop** address is not specified, the following rules apply to determine the source and destination IP addresses used in the OAM packet:

- If an IPv4 address is configured under the interface, then:

- OAM packet source IP address = Interface's IPv4 address
- OAM packet destination IP address = 127.0.0.0
- Else, if an IPv6 global address is configured under the interface, then:
 - OAM packet source IP address = Interface's IPv6 global address
 - OAM packet destination IP address = 0::ff:127.0.0.0
- Else, if an IPv6 link-local address is assigned to the interface, then:
 - OAM packet source IP address = Interface's IPv6 link-local address
 - OAM packet destination IP address = 0::ff:127.0.0.0

When the **next-hop {ipv4 | ipv6}** address is configured, the following rules apply to determine the source and destination IP addresses used in the OAM packet:

- If a next-hop IPv4 address is configured, then:
 - OAM packet source IP address = Interface's IPv4 address
 - OAM packet destination IP address = Configured next-hop IPv4 address



Note If there is no IPv4 address configured under the interface, then the delay-measurement probe does not send OAM packets.

- If a next-hop IPv6 address is configured, then:
 - OAM packet source IP address = Interface's IPv6 global address
 - OAM packet destination IP address = Configured next-hop IPv6 address



Note If there is no IPv6 global address configured under the interface, then the delay-measurement probe does not send OAM packets.

This example shows how to enable PM for link delay over an interface with IPv4 address configured:

```
interface TenGigE0/0/0/0
  ipv4 address 10.10.10.1 255.255.255.0

performance-measurement
  interface TenGigE0/0/0/0
    delay-measurement
```

This example shows how to enable PM for link delay over an interface IPv6 address configured:

```
interface TenGigE0/0/0/0
  ipv6 address 10:10:10::1/64
```

```
performance-measurement
interface TenGigE0/0/0/0
delay-measurement
```

This example shows how to enable PM for link delay over an interface with a specified next-hop IPv4 address:

```
interface TenGigE0/0/0/0
ipv4 address 10.10.10.1 255.255.255.0

performance-measurement
interface TenGigE0/0/0/0
next-hop ipv4 10.10.10.2
delay-measurement
```

This example shows how to enable PM for link delay over an interface with a specified next-hop IPv6 address:

```
interface TenGigE0/0/0/0
ipv6 address 10:10:10::1/64

performance-measurement
interface TenGigE0/0/0/0
next-hop ipv6 10:10:10::2
delay-measurement
```

This example shows how to enable PM for link delay over an interface with only IPv6 link-local address:

```
interface TenGigE0/0/0/0
ipv6 enable

performance-measurement
interface TenGigE0/0/0/0
delay-measurement
```

Verification

```
RP/0/0/CPU0:router# show performance-measurement profile default 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
  TX interval                 : 3300 (effective: 15000) uSec
  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 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 [<i>interface</i>]	Displays the PM link-delay probe history for interfaces.
show performance-measurement history aggregated interfaces [<i>interface</i>]	Displays the PM link-delay aggregated history for interfaces.
show performance-measurement history advertisement interfaces [<i>interface</i>]	Displays the PM link-delay advertisement history for interfaces.
show performance-measurement counters [<i>interface interface</i>] [<i>location location-name</i>]	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 [<i>location location-name</i>]	Displays the PM for link-delay summary on the remote-end router (responder).
show performance-measurement responder interfaces [<i>interface</i>]	Displays PM for link-delay for interfaces on the remote-end router.

Command	Description
show performance-measurement responder counters [<i>interface interface</i>] [<i>location location-name</i>]	Displays the PM link-delay session counters on the remote-end router.

Configure a Static Delay Value on an Interface

You can configure an interface to advertise a static delay value, instead of the measured delay value. When you configure a static delay value, the advertisement is triggered immediately. The average, minimum, and maximum advertised values will use the static delay value, with a variance of 0.

Scheduled probes will continue, and measured delay metrics will be aggregated and stored in history buffer. However, advertisement threshold checks are suppressed so that there are no advertisements of the actual measured delay values. If the configured static delay value is removed, the next scheduled advertisement threshold check will update the advertised measured delay values.

The static delay value can be configured from 1 to 16777215 microseconds (16.7 seconds).

This example shows how to configure a static delay of 1000 microseconds:

```
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-intf-dm)# advertise-delay 1000
```

Running Configuration

```
performance-measurement
 interface GigabitEthernet0/0/0/0
   delay-measurement
   advertise-delay 1000
 !
 !
 !
```

Verification

```
RP/0/RSP0/CPU0:ios# show performance-measurement interfaces detail
```

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

```
Interface Name: GigabitEthernet0/0/0/0 (ifh: 0x0)
Delay-Measurement           : Enabled
```

```
. . .
```

```
Last advertisement:
  Advertised at: Nov 29 2021 21:53:00.656 (7.940 seconds ago)
  Advertised reason: Advertise delay config
  Advertised delays (uSec): avg: 1000, min: 1000, max: 1000, variance: 0
```

```
. . .
```

SR Performance Measurement Named Profiles

CLI Changes for Segment Routing Performance Measurement

- Prior to Cisco IOS XR release 7.6.1, performance measurement delay and liveness named profiles can be configured using the **performance-measurement {delay-profile | liveness-profile} {sr-policy | endpoint | interface} name name** command.

Starting with Cisco IOS XR release 7.6.1, performance measurement delay and liveness named profiles can also be configured using the **performance-measurement {delay-profile | liveness-profile} name name** command. The configuration is stored in NVRAM in the same CLI format used to create it. There is no conversion from the old CLI to the new CLI.

Starting with Cisco IOS XR release 7.10.1, the **{sr-policy | endpoint | interface} name name** CLI has been deprecated. Old configurations stored in NVRAM will be rejected at boot-up.

As a result, performance measurement delay and liveness named profiles using the old CLI must be re-configured using the new CLI.



Note The default performance measurement delay and liveness profiles configured using the **performance-measurement {delay-profile | liveness-profile} {sr-policy | endpoint | interface} default** commands are still valid and are not affected.

- Prior to Cisco IOS XR release 7.6.1, performance measurement delay and liveness probes can be configured using the **burst-interval milliseconds** command.

Starting with Cisco IOS XR release 7.6.1, performance measurement delay and liveness probes can also be configured using the **tx-interval microseconds** command.

Starting with Cisco IOS XR release 7.10.1, the **burst-interval milliseconds** CLI has been deprecated. Old configurations stored in NVRAM will be rejected at boot-up.

As a result, performance measurement delay and liveness probes using the old CLI must be re-configured using the new CLI.

- In Cisco IOS XR release 7.10.1, the **measurement-mode loopback** CLI for delay and liveness profiles has been deprecated.

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 name profile2
Router(config-pm-dm-profile)# probe
Router(config-pm-dm-probe)# tx-interval 60000
Router(config-pm-dm-probe)# computation-interval 60
Router(config-pm-dm-probe)# protocol twamp-light
Router(config-pm-dm-probe)# tos dscp 63
Router(config-pm-dm-probe)# exit

Router(config-pm-dm-profile)# advertisement
Router(config-pm-dm-adv)# periodic
Router(config-pm-dm-adv-per)# interval 60
Router(config-pm-dm-adv-per)# minimum-change 1000
```

```
Router(config-pm-dm-adv-per)# threshold 20
Router(config-pm-dm-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
```

```
-----
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
TX interval                     : 3300 (effective: 15000) uSec
Packets per computation interval : 1000
Periodic advertisement
  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 name profile3
Router(config-pm-ld-profile)# probe
Router(config-pm-ld-probe)# tx-interval 60000
Router(config-pm-ld-profile)# probe
Router(config-pm-ld-probe)# tx-interval 60000
Router(config-pm-ld-probe)# tos dscp 10
Router(config-pm-ld-probe)# exit

Router(config-pm-ld-profile)# liveness-detection
Router(config-pm-ld-profile-ld)# multiplier 5
Router(config-pm-ld-profile-ld)# 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
```

```
-----
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
  TX interval                 : 3300 (effective: 15000) uSec
  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
  TX interval                 : 3300 (effective: 15000) uSec
  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

-----
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
  TX interval                 : 3300 (effective: 15000) uSec
  Destination sweeping mode  : Disabled
  Liveness detection parameters:
    Multiplier                : 3
    Logging state change     : Disabled
```

Delay Normalization

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
SR-TE Delay Normalization for OSPF	Release 7.3.1	This feature extends the current Delay Normalization feature to support OSPF.

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:

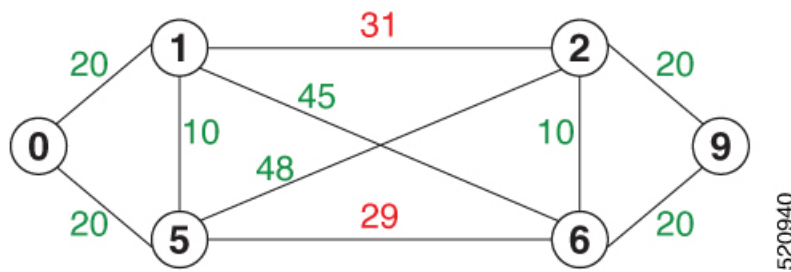
- **Dm** – measured Delay
- **Int** – configured normalized Interval

- **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.



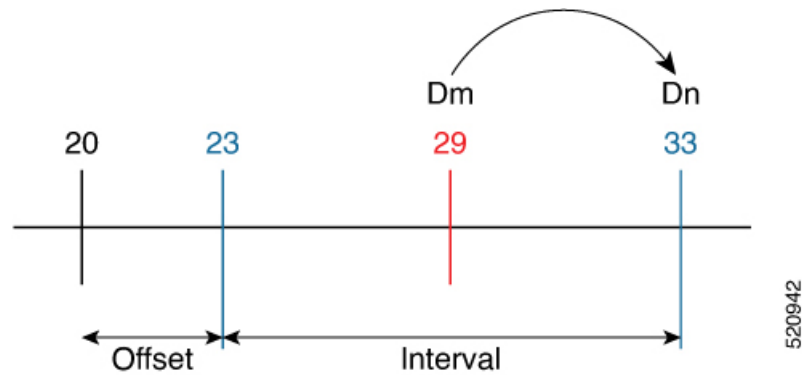
We can normalize the measured latency before it is advertised and used by IS-IS. Consider a scenario with the following:

- Interval = 10
- Offset = 3

The measured delays will be normalized as follows:

- $Dm = 29$
 $a = 29 / 10 = 2$ (2.9, rounded down to 2)
 $b = 2 * 10 + 3 = 23$

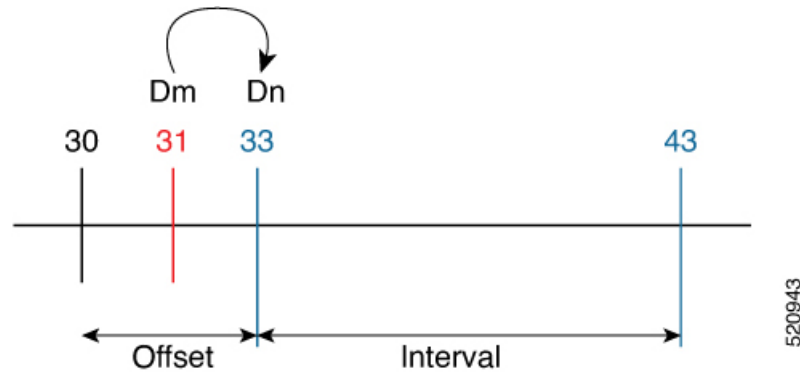
In this case, Dm (29) is greater than b (23); so Dn is equal to $b+I$ ($23 + 10$) = 33



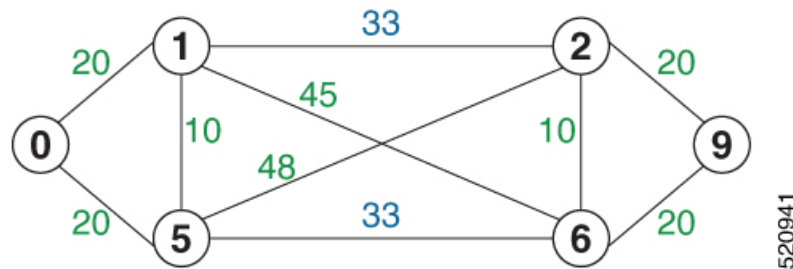
- $Dm = 31$
 $a = 31 / 10 = 3$ (3.1, rounded down to 3)

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

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 value 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 9: 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.

For information on configuring IGP penalty, see the following:

- [IS-IS Penalty for Link Delay Anomaly](#)
- [OSPF Penalty for Link Delay Anomaly](#)

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 anomaly 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
```

Delay Measurement for IP Endpoint

Table 10: Feature History Table

Feature Name	Release Information	Feature Description
IP Endpoint Delay Measurement 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). This feature is supported on IPv4, IPv6, and MPLS data planes.

Delay for an IP endpoint is the amount of time it takes for a data packet to travel from a source device to a specific IP endpoint within a network.

To measure a delay for a packet, also called a probe, is sent from a source device to the target IP endpoint.

The time from when the packet leaves the source to when it arrives at the endpoint is measured and recorded as the delay.

You can measure one-way delay, Two-way delay, and Roundtrip delay or delay in loop-back mode. For more information on Delay measurement, see Link Delay Measurement and Measurement Modes.

Collecting IP Endpoint Probe Statistics

- Statistics associated with the probe for delay metrics 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; that is, end of the periodic advertisement-interval (event: advertisement-interval expired)
 - Delay metrics last notified (event: notification-triggered)
- The following xpaths for MDT/EDT is 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

Guidelines and Limitations

You can specify a custom labeled path through one or more user-configured segment-lists. User-configured segment-list represents the forwarding path from sender to reflector when the probe is configured in delay-measurement mode.

- Examples of the custom segment-list include:
 - Probe in delay-measurement mode with a segment-list that includes Flex-Algo prefix SID of the endpoint
 - Probe in delay-measurement mode with a segment-list that includes a SID-list with labels to reach the endpoint or the sender (forward direction)
 - Probe in delay-measurement mode with a segment-list that includes BSID associated with SR policy to reach the end point.
- Endpoint segment list configuration is not supported under nondefault VRF.
- SR Performance Measurement endpoint session over BVI interface is not supported.

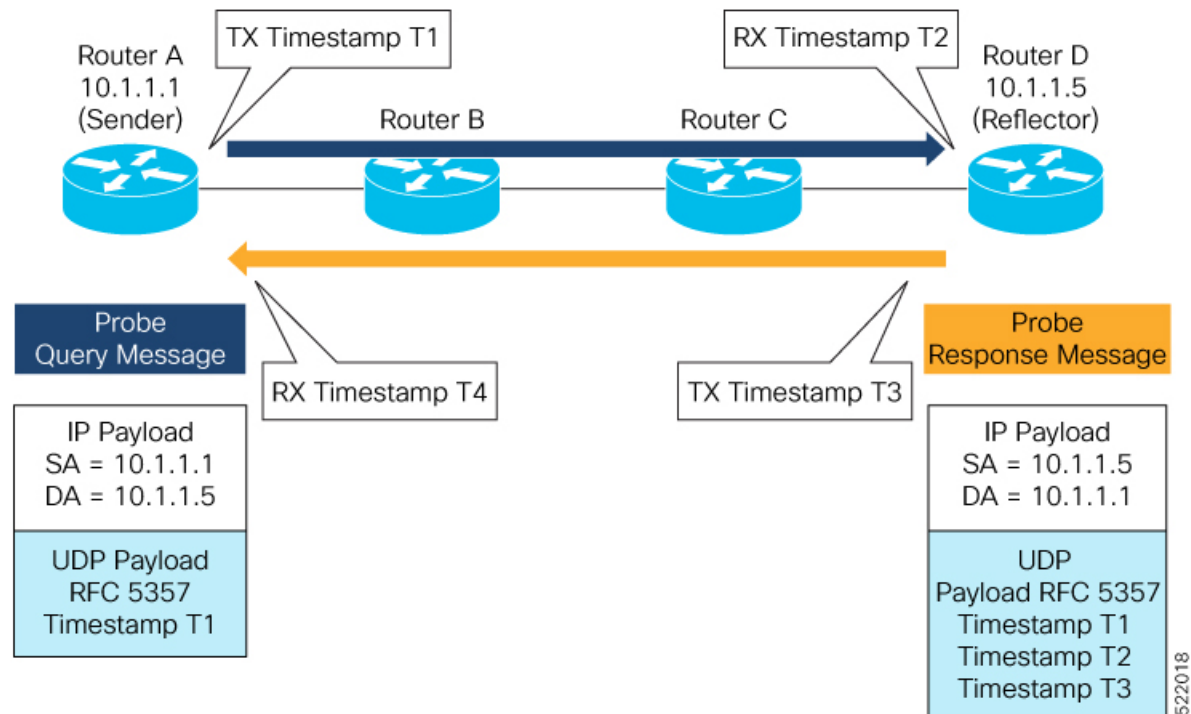
IP Endpoint Delay Measurement over MPLS Network Usecases

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



522018

Configuration

```
RouterA(config)# performance-measurement
RouterA(config-perf-meas)# endpoint ipv4 10.1.1.5
RouterA(config-pm-ep)# source-address ipv4 10.1.1.1
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 10.1.1.5
  source-address ipv4 10.1.1.1
  delay-measurement
  !
  !
 delay-profile endpoint default
  probe
  measurement-mode one-way
  !
  !
  !
```

Verification

```
RouterA# show performance-measurement endpoint ipv4 10.1.1.5
```

```
0/RSP0/CPU0
```

```
Endpoint name: IPv4-10.1.1.5-vrf-default
Source address      : 10.1.1.1
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 occurred

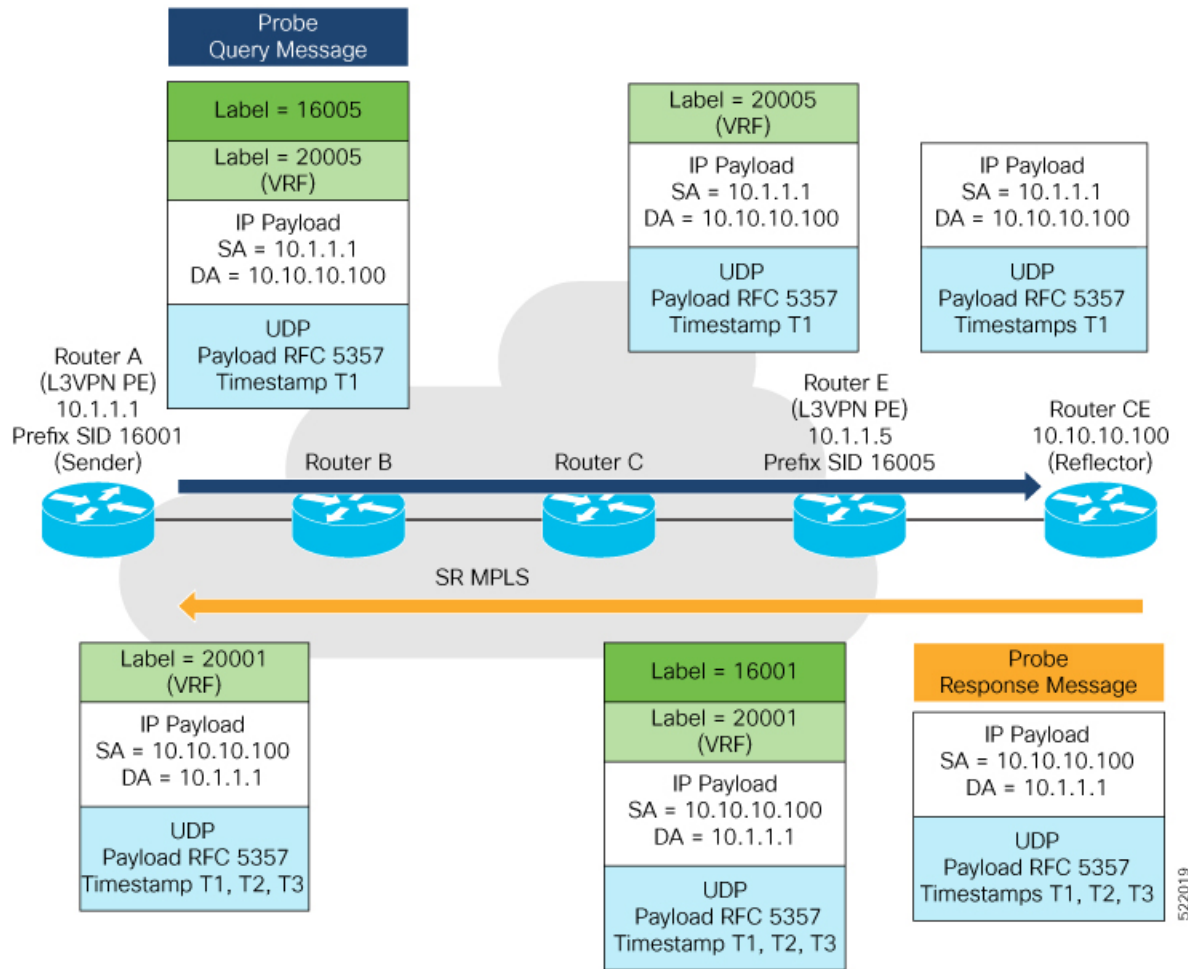
  Next advertisement:
    Threshold check scheduled in 4 more probes (roughly every 120 seconds)
    No probes completed

  Current computation:
    Started at: Jul 19 2021 16:28:06.723 (17.788 seconds ago)
    Packets 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 10.1.1.1
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 10.1.1.1
   delay-measurement
   !
 !
 delay-profile endpoint default
 probe
 measurement-mode one-way
    
```

522019

!
!
!

Verification

RouterA# `show performance-measurement endpoint vrf green`

0/RSP0/CPU0

```
Endpoint name: IPv4-10.10.10.100-vrf-green
Source address      : 10.1.1.1
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 occurred

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

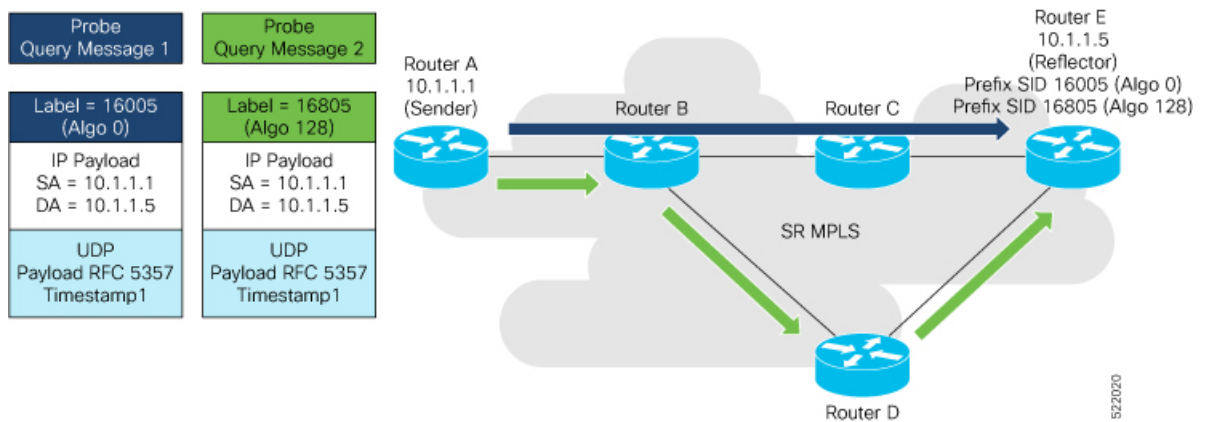
  Current computation:
    Not running: Unable to resolve (non-existing) vrf
```

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 10.1.1.5
RouterA(config-pm-ep)# source-address ipv4 10.1.1.1
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 10.1.1.5
  segment-list name SIDLIST1-Algo0
  !
  segment-list name SIDLIST2-FlexAlgo128
  !
  source-address ipv4 10.1.1.1
  delay-measurement
  !
  !
  delay-profile endpoint default
  probe
  measurement-mode one-way
  !
  !
  !
!
```

Verification

```
RouterA# show performance-measurement endpoint ipv4 10.1.1.5
```

```
0/RSP0/CPU0
```

```
Endpoint name: IPv4-10.1.1.5-vrf-default
```

```
Source address      : 10.1.1.1
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 occurred
```

```
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 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 occurred
```

```
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 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 occurred
```

```
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 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

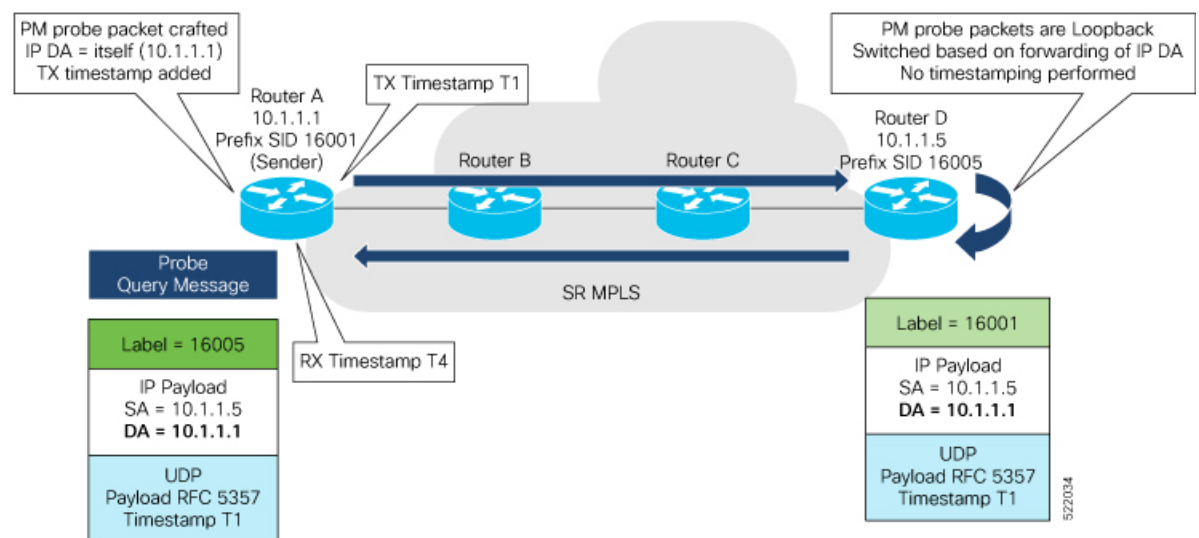
IP endpoint liveness detection leverages the loopback measurement-mode. The following workflow describes the sequence of events.

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 10.1.1.5
RouterA(config-pm-ep)# source-address ipv4 10.1.1.1
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
```

Running Configuration

```
performance-measurement
 endpoint ipv4 10.1.1.5
  source-address ipv4 10.1.1.1
  liveness-detection
  !
  !
  liveness-profile endpoint default
  liveness-detection
    multiplier 5
  !
!
!
end
```

Verification

```
RouterA# show performance-measurement endpoint ipv4 10.1.1.5
```

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

```
Endpoint name: IPv4-10.1.1.5-vrf-default
Source address      : 10.1.1.1
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

Table 11: Feature History Table

Feature Name	Release	Description
Segment Routing Performance Measurement for Link Delay and SR Policy Delay Using RFC 5357 (TWAMP Light) Encoding	Release 7.2.2	<p>This feature introduces support for Two-Way Active Measurement Protocol (TWAMP) Light (RFC 5357) for link delay and SR policy delay measurement. TWAMP Light adds two-way or round-trip measurement capabilities.</p> <p>Network performance data such as packet loss, delay and delay variation, and bandwidth utilization is a critical measure for Traffic Engineering (TE). This data provides service providers the characteristics of their networks for performance evaluation that is required to ensure the Service Level Agreements (SLAs). The performance measurement and delay variation feature allows you to measure those metrics and advertise them through IGP extensions as extended TE metrics.</p>

The PM for SR Policy 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 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.

Usage Guidelines and Limitations for PM for SR Policy Delay

The following usage guidelines and limitations apply:

- SR-PM delay measurement over SR Policy is supported on manually configured SR Policies and On-Demand SR Policies (ODN).
- SR-PM delay measurement over SR Policy is not supported on PCE-initiated SR Policies.
- Hardware clocks must be synchronized between the querier and the responder nodes of the link using PTP for one-way delay measurement.

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. Two-way delay and loopback modes are supported. See [Measurement Modes, on page 16](#) for more information.
- **tx-interval**: Interval for sending probe packet. The default value is 3000000 microseconds and the range is from 30000 to 15000000 microseconds.
- **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.
- **tos**: Type of Service
 - **dscp value**: The default value is 48 and the range is from 0 to 63.
 - **traffic-class value**: The default value is 6 and the range is from 0 to 7.
- **advertisement threshold-check**: minimum-delay/maximum-delay - 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**: Checks the minimum-delay metric change for threshold crossing for periodic advertisement. The default value 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**: Checks the minimum-delay metric change for threshold crossing for accelerated advertisement. 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 63
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 for delay.

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 for delay.

```

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
```

Verification

```
Router# show performance-measurement sr-policy
Mon Jan 20 18:48:41.002 PST
```

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

Policy Name	LSP ID	Tx/Rx	Avg/Min/Max/Variance
srte_c_10_ep_192.168.0.4	2	6/6	27012/26906/27203/106

```
Router# show performance-measurement sr-policy name srte_c_10_ep_192.168.0.4 detail verbose
Mon Jan 20 18:44:22.400 PST
```

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

```
SR Policy name: srte_c_10_ep_192.168.0.4
Color                : 10
Endpoint              : 192.168.0.4
Number of candidate-paths : 1

Candidate-Path:
Instance              : 2
Preference            : 100
Protocol-origin       : Configured
Discriminator         : 100
Source address        : 192.168.0.2
Reverse path label    : Not configured
Number of segment-lists : 1
Last advertisement:
  No advertisements have occurred
Next advertisement:
  Check scheduled at the end of the current probe (roughly every 30 seconds)
  Aggregated delays (uSec): avg: 45218, min: 26512, max: 82600, variance: 18706
  Rolling average (uSec): 45218
Last probe:
```



```

Packets 9, received: 9
Measured delays (uSec): avg: 45218, min: 26512, max: 82600, variance: 18706
Current Probe:
  Started at Jan 20 2020 18:44:19.170 (3.453 seconds ago)
  Packets 3, received: 3      Measured delays (uSec): avg: 26588, min: 26558, max:
26630, variance: 30
Next probe scheduled at Jan 20 2020 18:44:34.166 (in 11.543 seconds)
Next burst packet will be sent in 1.543 seconds
Burst packet sent every 5.0 seconds
Liveness Detection: Disabled

Segment-List          : R4
  16004
Number of atomic paths : 3
Last advertisement:
  No advertisements have occurred
Next advertisement:
  Aggregated delays (uSec): avg: 45218, min: 26512, max: 82600, variance: 18706
  Rolling average (uSec): 45218
Last probe:
  Packets 9, received: 9
  Measured delays (uSec): avg: 45218, min: 26512, max: 82600, variance: 18706
Current probe:
  Packets 3, received: 3
  Measured delays (uSec): avg: 26588, min: 26558, max: 26630, variance: 30
Liveness Detection: Disabled

Atomic path:
Hops          : 127.0.0.0
Session ID    : 33554434
Last advertisement:
  No advertisements have occurred
Next advertisement:
  Aggregated delays (uSec): avg: 45407, min: 26629, max: 82600, variance: 18778
  Rolling average (uSec): 45407
Last Probe:
  Packets 3, received: 3
  Measured delays (uSec): avg: 45407, min: 26629, max: 82600, variance: 18778
Current Probe:
  Packets 1, received: 1
  Measured delays (uSec): avg: 26630, min: 26630, max: 26630, variance: 0
Probe samples:
  Packet Rx Timestamp      Measured Delay (nsec)
  Jan 20 2020 18:44:19.198      26630730
Liveness Detection: Disabled

Atomic path:
Hops          : 127.0.0.1
Session ID    : 33554435
Last advertisement:
  No advertisements have occurred
Next advertisement:
  Aggregated delays (uSec): avg: 45128, min: 26521, max: 81961, variance: 18607
  Rolling average (uSec): 45128
Last Probe:
  Packets 3, received: 3
  Measured delays (uSec): avg: 45128, min: 26521, max: 81961, variance: 18607
Current Probe:
  Packets 1, received: 1
  Measured delays (uSec): avg: 26576, min: 26576, max: 26576, variance: 0
Probe samples:
  Packet Rx Timestamp      Measured Delay (nsec)
  Jan 20 2020 18:44:19.198      26576938
Liveness Detection: Disabled

```

```

Atomic path:
Hops           : 192.168.0.4
Session ID     : 33554433
Last advertisement:
  No advertisements have occurred
Next advertisement:
  Aggregated delays (uSec): avg: 45119, min: 26512, max: 81956, variance: 18607
  Rolling average (uSec): 45119
Last Probe:
  Packets 3, received: 3
  Measured delays (uSec): avg: 45119, min: 26512, max: 81956, variance: 18607
Current Probe:
  Packets 1, received: 1
  Measured delays (uSec): avg: 26558, min: 26558, max: 26558, variance: 0
Probe samples:
  Packet Rx Timestamp      Measured Delay (nsec)
  Jan 20 2020 18:44:19.198    26558375
Liveness Detection: Disabled

```

```

Router# show performance-measurement history probe sr-policy
Mon Jan 20 18:46:55.445 PST

```

```

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

```

```

SR Policy name: srte_c_10_ep_192.168.0.4
Color           : 10
Endpoint        : 192.168.0.4

```

```

Candidate-Path:
Preference      : 100
Protocol-origin : Configured
Discriminator   : 100

```

```

Delay-Measurement history (uSec):
  Probe Start Timestamp      Pkt (TX/RX)  Average      Min      Max
  Jan 20 2020 18:46:34.174    9/9          26880        26684    27070
  Jan 20 2020 18:46:19.174    9/9          26899        26822    27004
  Jan 20 2020 18:46:04.173    9/9          26813        26571    27164
  Jan 20 2020 18:45:49.172    9/9          26985        26713    27293
  Jan 20 2020 18:45:34.172    9/9          26744        26557    27005
  Jan 20 2020 18:45:19.171    9/9          26740        26435    27093
  Jan 20 2020 18:45:04.171    9/9          27115        26938    27591
  Jan 20 2020 18:44:49.171    9/9          26878        26539    27143
  Jan 20 2020 18:44:34.171    9/9          26824        26562    27265
  Jan 20 2020 18:44:19.170    9/9          26944        26558    27422
  Jan 20 2020 18:44:06.543    9/9          45218        26512    82600

```

```

Segment-List           : R4
16004

```

```

Delay-Measurement history (uSec):
  Probe Start Timestamp      Pkt (TX/RX)  Average      Min      Max
  Jan 20 2020 18:46:34.174    9/9          26880        26684    27070
  Jan 20 2020 18:46:19.174    9/9          26899        26822    27004
  Jan 20 2020 18:46:04.173    9/9          26813        26571    27164
  Jan 20 2020 18:45:49.172    9/9          26985        26713    27293
  Jan 20 2020 18:45:34.172    9/9          26744        26557    27005
  Jan 20 2020 18:45:19.171    9/9          26740        26435    27093
  Jan 20 2020 18:45:04.171    9/9          27115        26938    27591
  Jan 20 2020 18:44:49.171    9/9          26878        26539    27143
  Jan 20 2020 18:44:34.171    9/9          26824        26562    27265
  Jan 20 2020 18:44:19.170    9/9          26944        26558    27422
  Jan 20 2020 18:44:06.543    9/9          45218        26512    82600

```

```
Atomic path:
Hops          : 127.0.0.0
Delay-Measurement history (uSec):
Probe Start Timestamp      Pkt (TX/RX)  Average      Min      Max
Jan 20 2020 18:46:34.174   3/3         26927       26747    27070
Jan 20 2020 18:46:19.174   3/3         26982       26970    27004
Jan 20 2020 18:46:04.173   3/3         26895       26647    27164
Jan 20 2020 18:45:49.172   3/3         27054       26764    27293
Jan 20 2020 18:45:34.172   3/3         26801       26694    27005
Jan 20 2020 18:45:19.171   3/3         26807       26524    27093
Jan 20 2020 18:45:04.171   3/3         27226       26938    27591
Jan 20 2020 18:44:49.171   3/3         26976       26644    27143
Jan 20 2020 18:44:34.171   3/3         26880       26679    27265
Jan 20 2020 18:44:19.170   3/3         26994       26630    27422
Jan 20 2020 18:44:06.543   3/3         45407       26629    82600
```

```
Atomic path:
Hops          : 127.0.0.1
Delay-Measurement history (uSec):
Probe Start Timestamp      Pkt (TX/RX)  Average      Min      Max
Jan 20 2020 18:46:34.174   3/3         26865       26705    26988
Jan 20 2020 18:46:19.174   3/3         26846       26822    26881
Jan 20 2020 18:46:04.173   3/3         26787       26581    26939
Jan 20 2020 18:45:49.172   3/3         26954       26728    27180
Jan 20 2020 18:45:34.172   3/3         26724       26577    26957
Jan 20 2020 18:45:19.171   3/3         26705       26452    27032
Jan 20 2020 18:45:04.171   3/3         27043       26972    27124
Jan 20 2020 18:44:49.171   3/3         26848       26550    27062
Jan 20 2020 18:44:34.171   3/3         26800       26562    27204
Jan 20 2020 18:44:19.170   3/3         26927       26576    27327
Jan 20 2020 18:44:06.543   3/3         45128       26521    81961
```

```
Atomic path:
Hops          : 192.168.0.4
Delay-Measurement history (uSec):
Probe Start Timestamp      Pkt (TX/RX)  Average      Min      Max
Jan 20 2020 18:46:34.174   3/3         26848       26684    26967
Jan 20 2020 18:46:19.174   3/3         26871       26833    26913
Jan 20 2020 18:46:04.173   3/3         26759       26571    26876
Jan 20 2020 18:45:49.172   3/3         26947       26713    27163
Jan 20 2020 18:45:34.172   3/3         26708       26557    26939
Jan 20 2020 18:45:19.171   3/3         26708       26435    27075
Jan 20 2020 18:45:04.171   3/3         27078       27016    27138
Jan 20 2020 18:44:49.171   3/3         26812       26539    27043
Jan 20 2020 18:44:34.171   3/3         26793       26582    27181
Jan 20 2020 18:44:19.170   3/3         26911       26558    27308
Jan 20 2020 18:44:06.543   3/3         45119       26512    81956
```

```
Router# show performance-measurement counters sr-policy name srte_c_10_ep_192.168.0.4
Mon Jan 20 18:47:55.499 PST
```

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

```
SR Policy name: srte_c_10_ep_192.168.0.4
```

```
Candidate-Path:
```

```
Instance          : 2
Preference        : 100
Protocol-origin   : Configured
Discriminator     : 100
```

```
Packets:
```

```
Total sent          : 141
Total received      : 141
```

```
Errors:
  Total sent errors           : 0
  Total received errors      : 0
Probes:
  Total started              : 16
  Total completed            : 15
  Total incomplete           : 0
  Total advertisements       : 2
Segment-List                 : R4
16004
Packets:
  Total sent                  : 141
  Total received              : 141
Errors:
  Total sent errors           : 0
  Total received errors      : 0
Probes:
  Total started              : 16
  Total completed            : 15
  Total incomplete           : 0
  Total advertisements       : 2
```