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	
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.

- Liveness Monitoring, on page 2
- Delay Measurement, on page 27
- Fallback delay advertisement, on page 68
- Two-Way Active Measurement Protocol Light Source Address Filtering, on page 71
- Synthetic Loss Measurement, on page 75
- Delay and synthetic loss measurement for GRE tunnel interfaces, on page 81
- Delay Measurement Using Software Timestamp, on page 84

Liveness Monitoring

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. See *Configure PTP* in *System Management Configuration Guide* for more information on configuring PTP.

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 2: Feature History Table

Feature Name	Release Information	Feature Description
Liveness Monitoring for IP Endpoint over SRv6 Network	Release 24.2.11	This feature now extends support on the Cisco NCS 540 Series routers running on Cisco IOS XR7.

Feature Name	Release Information	Feature Description
Liveness Monitoring for IP Endpoint over SRv6 Network	Release 24.2.1	Introduced in this release on the following Cisco NCS 540 router variants running on Cisco IOS XR: • N540-ACC-SYS • N540X-ACC-SYS • N540-24Z8Q2C-SYS In Segment Routing over an IPv6 network (SRv6), you can keep track of the operational status of both the forward and reverse paths of a particular node or IP endpoint. You can use this information for troubleshooting, network maintenance, and optimizing network performance. Additionally, you can use flow labels to verify the liveness of each subsequent hop path toward the IP endpoint of that path. So that, when network traffic is distributed across multiple available paths towards an IP endpoint, liveness detection tracks the operational status of each of these paths towards the IP endpoint. The feature introduces these changes: **CLI:** • The **reverse-path** and **segment-list name** keywords are introduced in the **segment-routing traffic-eng explicit** command. • The **source-address ipv6** is introduced in the **performance-measurement endpoint** command. **YANG Data Model:** • `Cisco-IOS-XR-um-performance-measurement-cfg` • `Cisco-IOS-XR-perf-meas-oper.yang` (see GitHub, YANG Data Models Navigator)
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 SRV6, 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.
- When the endpoint is reachable using SRv6, the forwarding stage imposes the SRv6 encapsulation.

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.

- Reverse Path: To detect the liveness of the reverse of the segment, you can configure the reverse path using the **reverse-path** command.

  The default reverse path configured under the endpoint submode is only used for sessions with segment list. The endpoint session without a segment list does not support reverse path configuration and will not use this reverse path.

  The **reverse-path** under the **performance-measurement endpoint** is used as the default reverse path if there are no reverse paths configured under the segment list.

  Use the **reverse-path** under the **performance-measurement endpoint segment-routing traffic-eng explicit segment-list name** to configure the reverse path under segment list.

  The reverse type must be the same as the forward path. Using different types for forward and reverse paths is not supported. For example, uSID forward path and uSID reverse path; MPLS forward path and MPLS reverse path.

  User-configured segment-list can also represent the reverse path (reflector to sender) when probe is configured in liveness detection mode. Up to 128 segment-lists can be configured under a probe. An additional PM session is created for each segment-list. Segment-lists are configured under **segment-routing traffic-eng segment-list** submode. See SR-TE Policy with Explicit Path for details about configuring segment lists.

- Flow Label: The flow label field in the IPv6 header is used to carry information that helps distribute traffic across multiple network paths. The flow label is a 20-bit field in the IPv6 header designed to carry information about the flow of packets, which routers can use to identify and differentiate between different traffic flows. Flow label sweeping uses a flow label to distribute the traffic load across multiple paths to the endpoint.

  Use the **flow-label** keyword to configure flow label.

Usage Guidelines and Limitations

- Liveness session without segment list for an endpoint in a non-default VRF is not supported.
- Liveness on SRv6 is supported on Cisco NCS 5508 starting from Cisco IOS XR Release 24.3.1.
- IPv6 Endpoint Liveness in Default VRF over SRv6 is not supported.
- IP Endpoint in VRF without segment list, over SRv6 underlay (dynamic uDT6 or uDT4) 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.

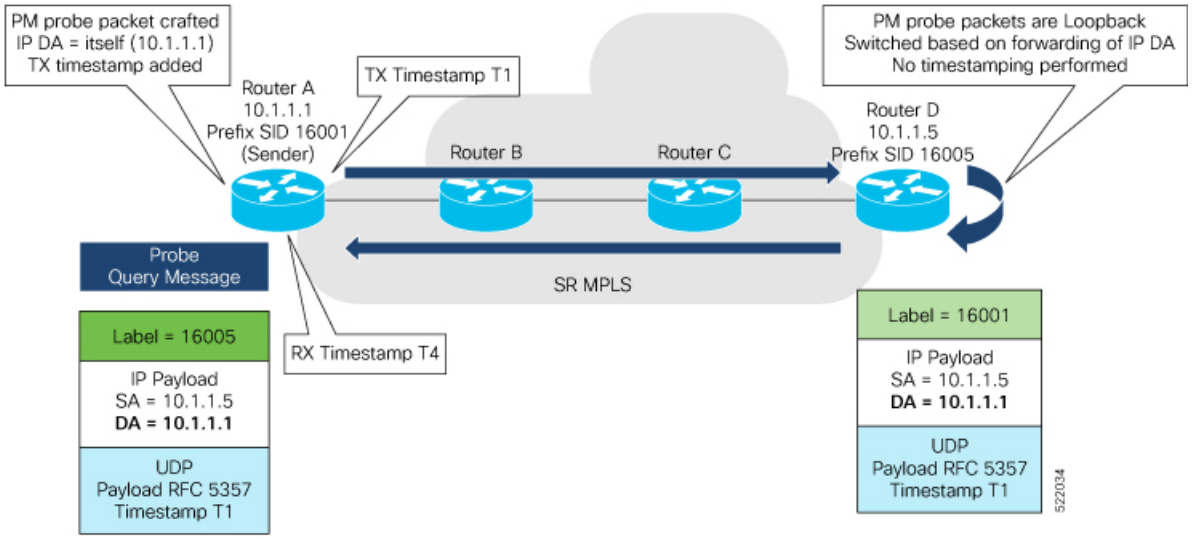
   The received timestamp (T4) stored.

   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
RouterA(config-pm-ld-ep)# probe
RouterA(config-pm-ld-ep-probe)# measurement-mode loopback
```

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
  !
  probe
   measurement-mode loopback
  !
 !
!
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
```

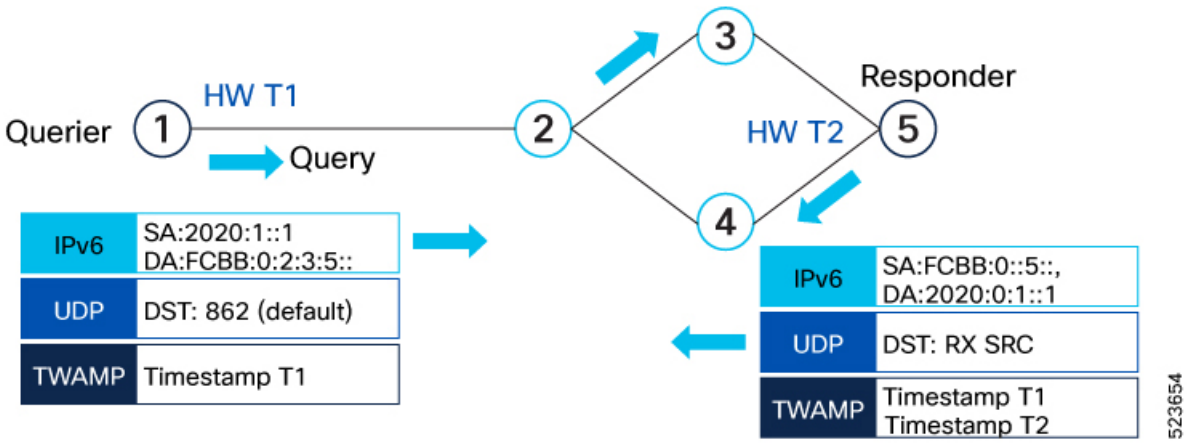
IP Endpoint Liveness in an SRv6 Network

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 and sets the IP destination address (DA) on the probe packets to the loopback value of the sender itself. The sender adds the transmit timestamp (T1) to the payload and encapsulates the probe packet with the SRv6 labels corresponding to the endpoint.
2. The network delivers the PM probe packets by following the IPv6 or segment lists towards the endpoint.
3. The endpoint receives the PM probe packets.

4. The sender node receives the PM probe packets and stores the received timestamp. If the sender node doesn't receive the specified number of probe packets based on the configured multiplier, it declares the PM session down.

Figure 2: IP Endpoint Liveness In an SRv6 Network



Note Liveness is not supported for non-default VRF.

Configuration Example

```
Router(config)#performance-measurement
Router(config-perf-meas)#source-address ipv6 2020:1::1
Router(config-perf-meas)#endpoint ipv6 FCBB:0::5::
Router(config-pm-ep)#exit
Router(config-perf-meas)#liveness-profile endpoint default
Router(config-pm-ld-ep)#probe
Router(config-pm-ld-ep-probe)#exit
Router(config-pm-ld-ep)#liveness-detection
Router(config-pm-ld-ep-ld)#multiplier 3
Router(config-pm-ld-ep-ld)#
```

The following example shows how to configure liveness with segment list and reverse path.

```
Router(config-sr)#traffic-eng
Router(config-sr-te)#segment-lists
Router(config-sr-te-segment-lists)#srv6
Router(config-sr-te-sl-global-srv6)#sid-format usid-f3216
Router(config-sr-te-sl-global-srv6)#exit
Router(config-sr-te-sl-global)#segment-list test
Router(config-sr-te-sl)#srv6
Router(config-sr-te-sl-srv6)#index 10 sid ff::2
Router(config-sr-te-sl-srv6)#index 20 sid ff::3
```

The following example shows how to configure liveness reverse path under segment list and under endpoint:

```
Router(config)#performance-measurement
Router(config-perf-meas)#endpoint ipv6 ff::2

/* Configure reverse path under segment list name *\
Router(config-pm-ep)#segment-routing traffic-eng explicit segment-list name fwd-path
```

```
Router(config-pm-ep-sl)#reverse-path segment-list name rev-path
Router(config-pm-ep-sl)#exit

/* Configure reverse path under performance measurement endpoint *\
Router(config-pm-ep)# segment-routing traffic-eng explicit reverse-path segment-list name
rev-path-name
```

The following example shows how to configure liveness with flow label:

```
Router(config-perf-meas)#liveness-profile endpoint default
Router(config-pm-ld-ep)#probe
Router(config-pm-ld-ep-probe)#flow-label from 1000 to 20000 increment 16
Router(config-pm-ld-ep-probe)#liveness-detection
Router(config-pm-ld-ep-ld)#multiplier 3
```

The following example shows how to configure liveness with flow label sweeping:

```
Router#configure
Router(config)#performance-measurement
Router(config-perf-meas)#liveness-profile name profile-sweeping
Router(config-pm-ld-profile)# flow-label from 1000 to 20000 increment 16
Routerconfig-pm-ld-profile)#commit
```

Verification

```
Router# show performance-measurement endpoint detail
Endpoint name: IPv6-FCBB:0::5:::-vrf-default
  Source address              : 2020:1::1
  VRF name                    : default
  Liveness Detection          : Enabled
  Profile Keys:
    Profile name              : default
    Profile type              : Endpoint Liveness Detection
  Segment-list                : None
  Liveness Detection session:
    Session ID                : 4109
    Flow-label                : 1000
    Session State: Up
    Last State Change Timestamp: Jan 23 2024 16:06:01.214
    Missed count: 0

  Liveness Detection session:
    Session ID                : 4110
    Flow-label                : 2000
    Session State: Up
    Last State Change Timestamp: Jan 23 2024 16:06:01.214
    Missed count: 0

  Segment-list                : test-dm-two-carrier-sl2
    FCBB:0::5:2:e004::/64
      Format: f3216
    FCBB:0::5:3:e000::/64
      Format: f3216
    FCBB:0::5:2:e004::/64
      Format: f3216
    FCBB:0::5:2:e000::/64
      Format: f3216
    FCBB:0::5:1:e000::/64
      Format: f3216
    FCBB:0::5:1:e004::/64
      Format: f3216
    FCBB:0::5:4:e000::/64
      Format: f3216
    FCBB:0::5:4::/48
```

```
    Format: f3216
Liveness Detection session:
  Session ID               : 4111
  Flow-label               : 1000
  Session State: Up
  Last State Change Timestamp: Jan 23 2024 16:06:01.217
  Missed count: 0

Liveness Detection session:
  Session ID               : 4112
  Flow-label               : 2000
  Session State: Up
  Last State Change Timestamp: Jan 23 2024 16:06:01.217
  Missed count: 0
```

SR Policy Liveness Monitoring

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
SR Policy Liveness Monitoring on Segment Routing over IPv6 (SRv6)	Release 7.11.1	In segment routing over IPv6 (SRv6), you can now verify end-to-end traffic forwarding over an SR policy candidate path by periodically sending probe messages. Performance monitoring on an SRv6 network enables you to track and monitor traffic flows at a granular level. Earlier releases supported SR policy liveness monitoring over an SR policy candidate path on MPLS.

Feature Name	Release Information	Feature Description
SR Performance Measurement Named Profiles	Release 7.3.1	You can use this feature to create specific performance measurement delay and liveness profiles, and associate it with an SR policy. 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. The **performance-measurement delay-profile sr-policy** command was updated with the **name** *profile* keyword-argument combination. The **performance-measurement liveness-profile sr-policy** command was updated with the **name** *profile* keyword-argument combination. The **performance-measurement delay-measurement** command was updated with **delay-profile name** *profile*. The **performance-measurement liveness-detection** command was updated with **liveness-profile name** *profile*
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:

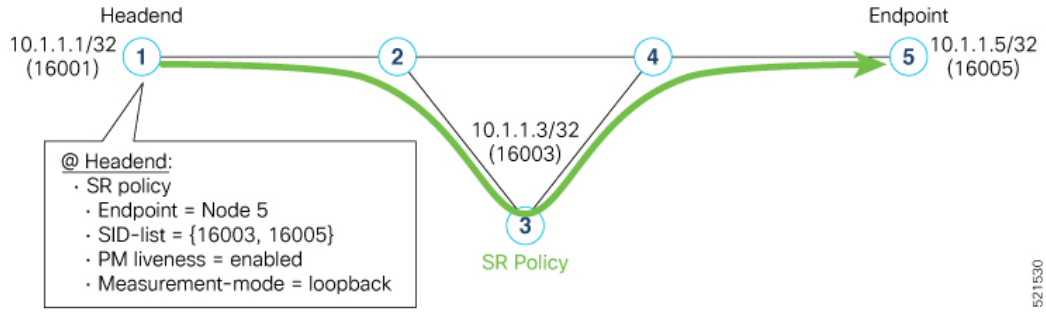
- Allows both liveness monitoring and delay measurement using a single-set of PM packets as opposed to running separate monitoring sessions for each purpose. This improves the overall scale by reducing the number of PM sessions required.
- Eliminates network and device complexity by reducing the number 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.

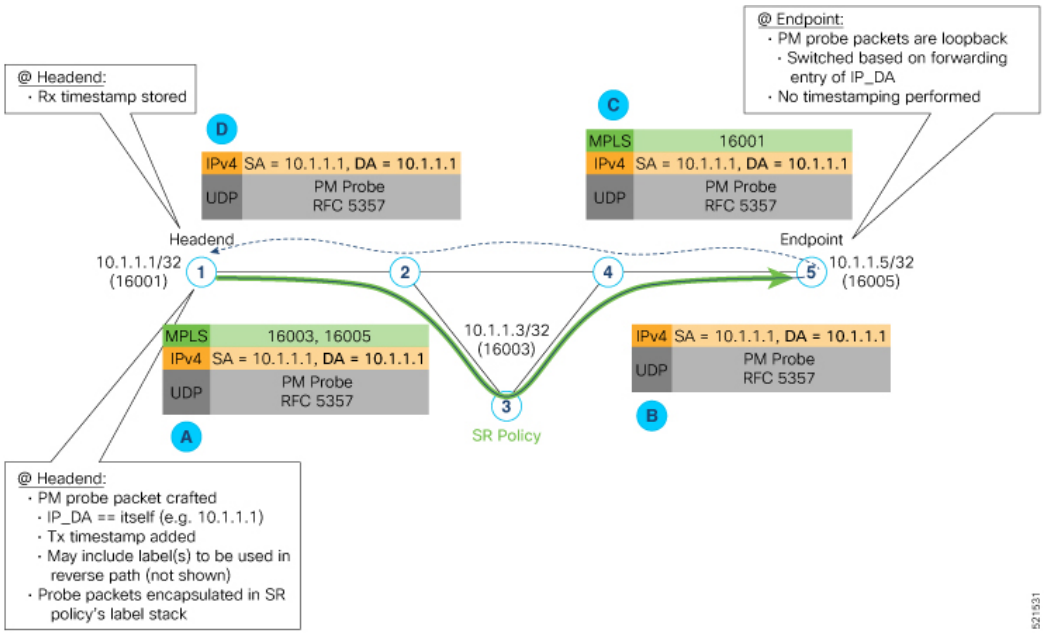
  If the head-end node inserted label(s) for the reverse path, then the packets are switched back at the end-point node based on the forwarding entry associated with the top-most reverse path label.

- **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 trigerred.



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.
- The head-end router doesn’t load-balance the liveness probes across bundle member links.

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

- **probe**: Configure the probe parameters.

- **measurement-mode**: Liveness detection must use loopback mode (see Measurement Modes, on page 27).
- **burst interval**: Interval for sending probe packet. The default value is 3000 milliseconds and the range is from 30 to 15000 milliseconds.
- **tos dscp** *value*: The default value is 48 and the range is from 0 to 63. You can modify the DSCP value of the probe packets, and use this value to priortize the probe packets from headend to tailend.
- **sweep destination ipv4 127.*x.x.x* range** *range*: Configure SR Policy ECMP IP-hashing mode. Specifiy the number of IP addresses to sweep. The range is from 0 (default, no sweeping) to 128. The option is applicable to IPv4 packets.

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

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

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

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

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

Enabling Liveness Monitoring under SR Policy

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

- **liveness-detection**: Enables end-to-end SR Policy Liveness Detection for all segment-lists of the active and standby candidate-path that are in the forwarding table.
- **liveness-profile name** *name*: Specifies the profile name for named profiles.
- **invalidation-action {down | none}**:
  - **Down (default)**: When the PM liveness session goes down, the candidate path is immediately operationally brought down.
  - **None**: When the PM liveness session goes down, no action is taken. If logging is enabled, the failure is logged but the SR Policy operational state 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)# measurement-mode loopback
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# burst-interval 1500
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# tos dscp 52
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# exit
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy)# liveness-detection
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-ld)# multiplier 5
```

Running Configuration:

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

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

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

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

Running Configuration:

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

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

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

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

Running Configuration

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

Enable Liveness Monitoring Under SR Policy

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

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

Running Config

```
segment-routing
 traffic-eng
```

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

Enable Liveness Monitoring under SR Policy with Optional Parameters

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

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

Running Config

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

Configure Segment Lists to Activate Candidate Paths in SRv6 for PM Liveness

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
Configure Segment Lists to Activate Candidate Paths in SRv6 for PM Liveness	Release 7.11.1	You can now enable a candidate path to be up by configuring the minimum number of active segment lists associated with the candidate path. The head-end router determines that a candidate path is up based on the minimum number of active segment lists configured. In earlier releases, the head-end router identified a candidate path as up only when all the segment lists associated with the path were active. The feature introduces these changes: **CLI:** • The **validation-cp minimum-active segment-lists** option is introduced in the **performance-measurement liveness-detection** command. **YANG Data Models:** • `Cisco-IOS-XR-infra-xtc-agent-cfg.yang` See (GitHub, Yang Data Models Navigator)

The state of the segment lists in a candidate path determines whether a candidate path is up or down. You can now configure the minimum number of active segment lists associated with a candidate path. The head-end router identifies a candidate path as up when one or more segment lists are active.

Note If the configured minimum number of active segment lists is greater than the number of available segment lists in a candidate path, the head-end router determines the candidate path as up only when all the segment lists are active.

In earlier releases, the router identified a candidate path as up only when all the segment lists associated with the path were active.

Configuration Example

Configure the minimum number of segment lists in SRv6

Perform this task to activate three segment lists to have the PM liveness session up:

```
Router(config)#segment-routing
Router(config-sr)#traffic-eng
Router(config-sr-te)#policy po-103
Router(config-sr-te-policy)#performance-measurement
Router(config-sr-te-policy-perf-meas)#liveness-detection
Router(config-sr-te-policy-live-detect)#validation-cp minimum-active segment-lists 3
```

Show Running Configuration

```
segment-routing
 traffic-eng
  policy po-103
   performance-measurement
    liveness-detection
     validation-cp minimum-active segment-lists 3
    !
   !
  !
 !
!
```

Verification

The following example shows three active segment-lists to have the PM liveness session up:

```
Router#show performance-measurement sr-policy liveness color 103 detail verbose private
Mon Oct 30 15:10:51.863 EDT

------------------------------------------------------------------------------------------------------------------------
0/1/CPU0
------------------------------------------------------------------------------------------------------------------------

SR Policy name: srte_c_103_ep_3::1
  Color                       : 103
  SRv6 Encap Source Address   : 1::1
  Endpoint                    : 3::1
  Handle                      : 0x00000000
  Policy to be deleted        : False
  Number of candidate-paths   : 1

  Candidate-Path:
    Instance                  : 5
    Preference                : 300
    Protocol-origin           : Configured
    Discriminator             : 300
    Profile Keys:
      Profile name            : default
      Profile type            : SR Policy Liveness Detection
    Candidate path to be deleted: False
    Source address            : 1::1
    Local label               : Not set
    Fast notification for session down: Disabled
      No fast notifications have been sent
    Number of segment-lists   : 3
    Liveness Detection: Enabled
      Minumum SL Up Required: 1
      Session State: Up
      Last State Change Timestamp: Oct 30 2023 15:10:16.322
```

```
  Missed count: 0

Segment-List              : sl-1041
  fccc:cc00:1:fe10:: (Local Adjacency SID)
  fccc:cc00:2:fe41::/64
    Format: f3216
  Segment List ID: 0
  Reverse path segment-List: Not configured
  Segment-list to be deleted: False
  Number of atomic paths  : 1
  Liveness Detection: Enabled
    Session State: Up
    Last State Change Timestamp: Oct 30 2023 15:10:16.322
    Missed count: 0

  Atomic path:
    Flow Label            : 0
    Session ID            : 4198
    Trace ID              : 738913600
    Atomic path to be deleted: False
    NPU Offloaded session : False
    Timestamping Enabled  : True
    Liveness Detection: Enabled
      Session State: Up
      Last State Change Timestamp: Oct 30 2023 15:10:16.322
      Missed count: 0
    Responder IP          : 1::1
    Number of Hops        : 3

Segment-List              : sl-1042
  fccc:cc00:1:fe10:: (Local Adjacency SID)
  fccc:cc00:2:fe42::/64
    Format: f3216
  Segment List ID: 0
  Reverse path segment-List: Not configured
  Segment-list to be deleted: False
  Number of atomic paths  : 1
  Liveness Detection: Enabled
    Session State: Up
    Last State Change Timestamp: Oct 30 2023 15:10:16.322
    Missed count: 0

  Atomic path:
    Flow Label            : 0
    Session ID            : 4199
    Trace ID              : 954039677
    Atomic path to be deleted: False
    NPU Offloaded session : False
    Timestamping Enabled  : True
    Liveness Detection: Enabled
      Session State: Up
      Last State Change Timestamp: Oct 30 2023 15:10:16.322
      Missed count: 0
    Responder IP          : 1::1
    Number of Hops        : 3

Segment-List              : sl-1043
  fccc:cc00:1:fe10:: (Local Adjacency SID)
  fccc:cc00:2:fe43::/64
    Format: f3216
  Segment List ID: 0
  Reverse path segment-List: Not configured
  Segment-list to be deleted: False
  Number of atomic paths  : 1
```

```
Liveness Detection: Enabled
  Session State: Up
  Last State Change Timestamp: Oct 30 2023 15:10:16.322
  Missed count: 0

Atomic path:
  Flow Label            : 0
  Session ID            : 4200
  Trace ID              : 1119107116
  Atomic path to be deleted: False
  NPU Offloaded session : False
  Timestamping Enabled  : True
  Liveness Detection: Enabled
    Session State: Up
    Last State Change Timestamp: Oct 30 2023 15:10:16.322
    Missed count: 0
  Responder IP          : 1::1
  Number of Hops        : 3

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

Configure Flow Labels in SRv6 Header for PM Liveness

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
Configure Flow Labels in SRv6 Header for PM Liveness	Release 7.11.1	You can now monitor the activeness of multiple paths for a given segment list using flow labels in the SRv6 header. In earlier releases, the SRv6 header didn't include flow labels. The feature introduces these changes: **CLI:** • The **flow-label** keyword is introduced in the **performance-measurement liveness-profile**command. **YANG Data Models:** • Cisco-IOS-XR-perfmeas-measurement-cfg.yang • Cisco-IOS-XR-perf-meas-oper.yang See (GitHub, Yang Data Models Navigator)

To monitor the activeness of multiple paths for a given segment list, you can configure the SRv6 header to include flow labels as the packet travels in the network. When there are multiple paths, different traffic flows

may use different paths. A flow label is a flow identifier and you can use different flow labels to monitor different ECMP paths. It's only used for IPv6 probe packets. Flow labels are 20-bit fields in the SRv6 header.

Configure flow labels in the SRv6 header

Perform the following task in the global configuration mode to configure flow labels in the SRv6 header:

```
Router#configure
Router(config)#performance-measurement
Router(config-perf-meas)#liveness-profile name name1
Router(config-pm-ld-profile)#probe flow-label from 0 to 1000000 increment 10
```

Running Configuration

```
performance-measurement
 liveness-profile name name1
  probe
   flow-label from 0 to 1000000 increment 10
  !
 !
!
```

Verification

The following example shows an SR-policy configured with flow labels:

```
Router#show performance-measurement sr-policy liveness color 1001 detail verbose private

Mon Oct 30 15:25:55.241 EDT

------------------------------------------------------------------------------------------------------------------------
0/1/CPU0
------------------------------------------------------------------------------------------------------------------------

SR Policy name: srte_c_1001_ep_3::1
  Color                       : 1001
  SRv6 Encap Source Address   : 1::1
  Endpoint                    : 3::1
  Handle                      : 0x00000000
  Policy to be deleted        : False
  Number of candidate-paths   : 1

  Candidate-Path:
    Instance                  : 3
    Preference                : 300
    Protocol-origin           : Configured
    Discriminator             : 300
    Profile Keys:
      Profile name            : profile-scale
      Profile type            : Generic Liveness Detection
    Candidate path to be deleted: False
    Source address            : 1::1
    Local label               : Not set
    Fast notification for session down: Disabled
      No fast notifications have been sent
    Number of segment-lists   : 2
    Liveness Detection: Enabled
      Minumum SL Up Required: 2
      Session State: Up
      Last State Change Timestamp: Oct 26 2023 15:31:43.478
      Missed count: 0

    Segment-List              : sl-1041
```

```
  fccc:cc00:1:fe10:: (Local Adjacency SID)
  fccc:cc00:2:fe41::/64
    Format: f3216
  Segment List ID: 0
  Reverse path segment-List: Not configured
  Segment-list to be deleted: False
  Number of atomic paths  : 2
  Liveness Detection: Enabled
    Session State: Up
    Last State Change Timestamp: Oct 26 2023 15:31:43.478
    Missed count: 0

  Atomic path:
    Flow Label            : 0
    Session ID            : 4178
    Trace ID              : 280178832
    Atomic path to be deleted: False
    NPU Offloaded session : False
    Timestamping Enabled  : True
    Liveness Detection: Enabled
      Session State: Up
      Last State Change Timestamp: Oct 26 2023 15:31:43.478
      Missed count: 0
    Responder IP          : 1::1
    Number of Hops        : 3

  Atomic path:
    Flow Label            : 10
    Session ID            : 4179
    Trace ID              : 1866227171
    Atomic path to be deleted: False
    NPU Offloaded session : False
    Timestamping Enabled  : True
    Liveness Detection: Enabled
      Session State: Up
      Last State Change Timestamp: Oct 26 2023 15:31:43.478
      Missed count: 0
    Responder IP          : 1::1
    Number of Hops        : 3

Segment-List              : sl-scale
  fccc:cc00:1:fe10:: (Local Adjacency SID)
  fccc:cc00:2:fed1::/64
    Format: f3216
  Segment List ID: 0
  Reverse path segment-List: Not configured
  Segment-list to be deleted: False
  Number of atomic paths  : 2
  Liveness Detection: Enabled
    Session State: Up
    Last State Change Timestamp: Oct 26 2023 15:31:43.478
    Missed count: 0

  Atomic path:
   Flow Label            : 0
    Session ID            : 4180
    Trace ID              : 2609815826
    Atomic path to be deleted: False
    NPU Offloaded session : False
    Timestamping Enabled  : True
    Liveness Detection: Enabled
      Session State: Up
      Last State Change Timestamp: Oct 26 2023 15:31:43.478
      Missed count: 0
```

```
    Responder IP          : 1::1
    Number of Hops        : 3

  Atomic path:
    Flow Label            : 10
    Session ID            : 4181
    Trace ID              : 170501506
    Atomic path to be deleted: False
    NPU Offloaded session : False
    Timestamping Enabled  : True
    Liveness Detection: Enabled
      Session State: Up
      Last State Change Timestamp: Oct 26 2023 15:31:43.478
      Missed count: 0
    Responder IP          : 1::1
    Number of Hops        : 3

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

Hardware offload of SR policy liveness monitoring

Table 6: Feature History Table

Feature Name	Release	Description
Hardware Offload of MPLS Liveness Monitoring	Release 7.10.1	You can now offload MPLS liveness monitoring for performance measurement to the router's hardware, which is the Network Processing Unit (NPU). This hardware-based approach improves efficiency and scalability, helping you meet delay-sensitive Service Level Agreements (SLAs). Previously, this monitoring was handled in software. The feature introduces a new keyword **npu-offload** under the **performance-measurement liveness-profile name***liveness profile* command.

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) at the router 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.

Note For SRv6 Liveness Monitoring - Hardware Offloading configuration, SRv6 uSID must be enabled. Use the following configuartion to enable SRv6 uSID must be enabled

```
Router(config)# hw-module profile segment-routing srv6 mode micro-segment format f3216
```

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

Note The configuration also applies to the default liveness profile and the configuration applies to the router.

```
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)#commit
```

Running Configuration

The running configuration for this feature is as shown for both SRv6 and MPLS:

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

Verification

Use the show command to verify the running configuration as shown for both MPLS and SRv6 Liveness Monitoring - Hardware Offloading :

- **Example for MPLS**

```
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
```

- **Example for SRv6**

```
Transport type              : Endpoint
Measurement type            : Liveness Detection
Endpoint name               : IPv6-fccc:ccc1:4::-vrf-default
endpoint                    : fccc:ccc1:4::
source                      : 192::2
vrf                         : default
Segment-list                : test-sl1
Liveness Detection: Enabled
```

```
  Session ID          : 3
  Profile Keys:
    Profile name        : default
    Profile type        : Endpoint Liveness Detection
  Session State: Down
  Missed count: 3
NPU Offloaded session       : True
  NPU number                : 0
  NPU session state         : Session created
    Retry 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 measuremnt uses the IP/UDP packet format defined in 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 31 Use to monitor delay experienced by data packets in a single link or path between two nodes in a network.
- IP endpoint delay measurement: 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 61: 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 that are supported in SR PM.

Table 7: Measurement Mode Requirements

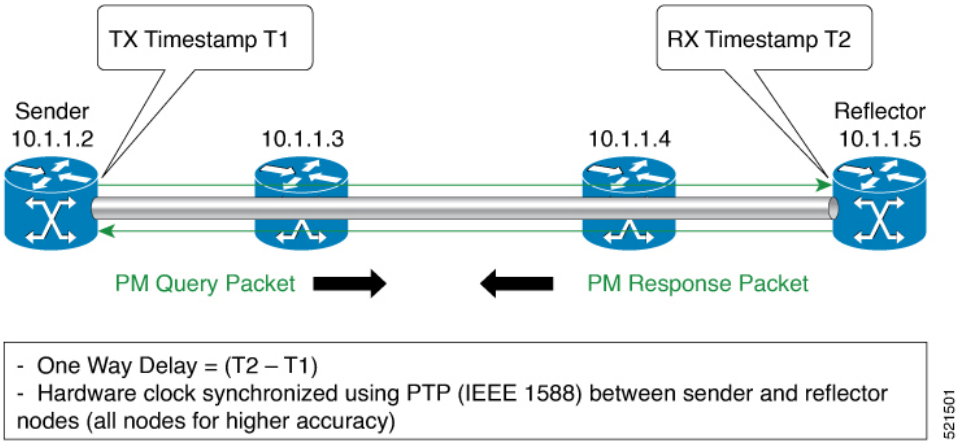
Measurement Mode	Sender: PTP-Capable HW and HW Timestamping	Reflector: PTP-Capable HW and HW Timestamping	PTP Clock Synchronization between Sender and Reflector
One-way	Required	Required	Required
Two-way	Required	Required	Not Required
Loopback	Required	Not Required	Not Required

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 3: One-Way



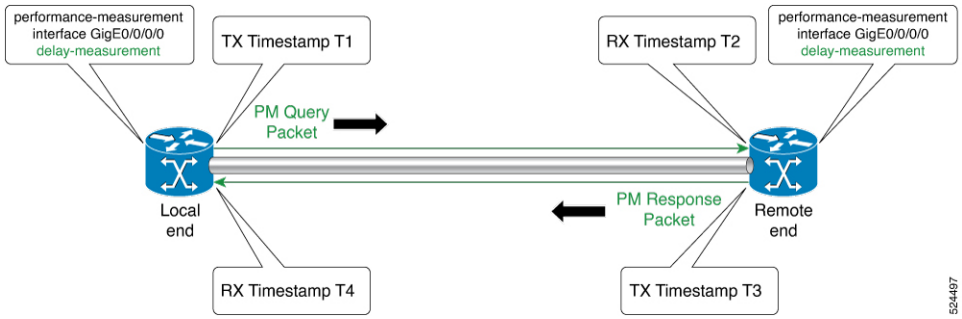
The PM query and response for one-way delay measurement can be described in the following steps:

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

Far-end delay metrics in one-way measurement mode

Table 8: Feature History Table

Feature Name	Release Information	Description
Far-end delay metrics in one-way measurement mode	Release 24.4.1	Segment Routing Performance Monitoring (SR PM) now enables network operators to compute both far-end (T4 – T3) and near-end (T2 – T1) delay metrics, offering a comprehensive view of end-to-end delay across the data path. Measuring far-end delay, from the responder to the querier node, enhances visibility and allows operators to precisely monitor and assess network performance. Previously, you could measure the near-end delay metrics for a given data path.



The far-end delay metric measures the round-trip time a packet takes to travel from the source to the destination across the Segment Routing network. It calculates the time it takes for the packet to reach the far end of a segment and return to the source.

Starting from Cisco IOS XR Release 24.4.1, the PM session automatically calculates both near-end (T2 – T1) and far-end (T4 – T3) metrics. The far-end delay is the delay from the responder to the querier node. This metric is automatically included in the CLI and telemetry outputs.

The far-end metrics provide network operators a complete view of delay across the entire data path, and not just the near-end. The feature enables network operators to understand the full path latency and ensures accurate network monitoring.

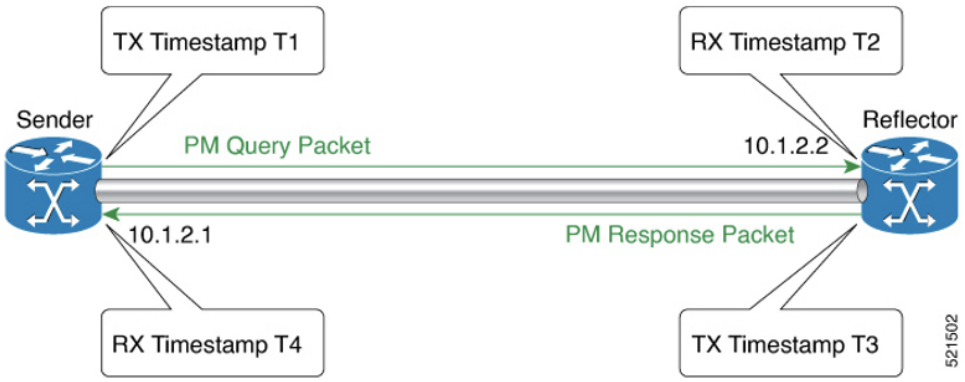
Note If the measurement mode is not set to one-way or is set to loopback, SR PM skips the far-end delay metric computation.

Two-Way Measurement Mode

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

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

Figure 4: Two-Way



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

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

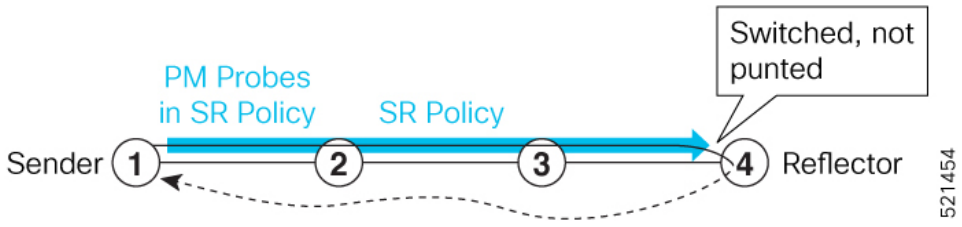
Loopback Measurement Mode

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

Delay measurements in Loopback mode are calculated as follows:

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

Figure 5: 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, routers support the following features:

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

Table 9: 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 27 for more information.
- **protocol**: Interface delay measurement using RFC 5357 with IP/UDP encap (TWAMP-Light).
- **burst interval**: Interval for sending probe packet. The default value is 3000 milliseconds and the range is from 30 to 15000 milliseconds.
- **computation interval**: Interval for metric computation. Default is 30 seconds; range is 1 to 3600 seconds.
- **periodic advertisement**: Periodic advertisement is enabled by default.
- **periodic-advertisement interval**: The default value is 120 seconds and the interval range is from 30 to 3600 seconds.
- **periodic-advertisement threshold**: 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
RP/0/0/CPU0:router(config-pm-dm-intf)# probe
RP/0/0/CPU0:router(config-pm-dm-intf-probe)# measurement-mode one-way
RP/0/0/CPU0:router(config-pm-dm-intf-probe)# burst-interval 60
RP/0/0/CPU0:router(config-pm-dm-intf-probe)# 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-proto-mode)# querier-dst-port 12000
```

Enable PM for Link Delay Over an Interface

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

```
RP/0/0/CPU0:router(config)# performance-measurement
RP/0/0/CPU0:router(config-perf-meas)# interface TenGigE0/0/0/0
RP/0/0/CPU0:router(config-pm-intf)# 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 interface
Thu Dec 12 14:13:16.029 PST

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

```
    Threshold crossing check                    : Minimum-delay

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

Thu Dec 12 14:09:59.162 PST

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


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

SR Policy Delay-Measurement:
  Total active sessions                         : 0
  Counters:
    Packets:
      Total sent                                : 0
      Total received                            : 0
    Errors:
        TX:
          Reason interface down                 : 0
          Reason no MPLS caps                   : 0
          Reason no IP address                  : 0
          Reason other                          : 0
        RX:
          Reason negative delay                 : 0
          Reason delay threshold exceeded       : 0
          Reason missing TX timestamp           : 0
          Reason missing RX timestamp           : 0
```

```
          Reason probe full                    : 0
          Reason probe not started             : 0
          Reason control code error            : 0
          Reason control code notif            : 0
    Probes:
      Total started                            : 0
      Total completed                          : 0
      Total incomplete                         : 0
      Total advertisements                     : 0

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

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

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

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

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

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

```
State                              : Up

Delay Measurement session:
  Session ID          : 1

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

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

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

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

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

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

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

Command	Description
show performance-measurement responder counters [**interface** *interface*] [**location** *location-name*]	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

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

Delay Profile

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

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

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

Apply the delay profile for an SR Policy.

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

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

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

Running Configuration

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

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

Verification

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

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

```
----------------------------
SR Policy Delay Measurement Profile Name: profile2
  Profile configuration:
    Measurement mode                          : One-way
    Protocol type                             : TWAMP-light
    Encap mode                                : UDP
    Type of service:
      PM-MPLS traffic class                   : 6
      TWAMP-light DSCP                        : 63
    Probe computation interval                : 60 (effective: 60) seconds
    Burst interval                            : 60 (effective: 60) mSec
    Packets per computation interval          : 1000
    Periodic advertisement                    : Enabled
      Interval                                : 60 (effective: 60) sec
      Threshold                               : 20%
      Minimum-change                          : 1000 uSec
    Advertisement accelerated                 : Disabled
    Advertisement logging:
      Delay exceeded                          : Disabled (default)
    Threshold crossing check                  : Maximum-delay
    Router alert                              : Disabled (default)
    Destination sweeping mode                 : Disabled
    Liveness detection parameters:
      Multiplier                              : 3
      Logging state change                    : Disabled
```

On-Demand SR Policy

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

Running Configuration

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

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

Liveness Profile

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

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

Apply the Liveness Profile for the SR Policy

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

For the same policy, you cannot enable delay-measurement (delay-profile) and liveness-detection (liveness-profile) at the same time. For example, if delay measurement is enabled, use the **no**

delay-measurement command to disable it, and then enable the following command for enabling liveness detection.

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

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

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

Running Configuration

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

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

Verification

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

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

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

SR Policy Liveness Detection Profile Name: profile3
  Profile configuration:
    Measurement mode                          : Loopback
    Protocol type                             : TWAMP-light
    Type of service:
```

```
      TWAMP-light DSCP                          : 10
    Burst interval                              : 60 (effective: 60) mSec
    Destination sweeping mode                   : Disabled
    Liveness detection parameters:
      Multiplier                                : 3
      Logging state change                      : Disabled
```

On-Demand SR Policy

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

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

Running Configuration

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

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

Verification

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

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

Delay Normalization

Table 10: 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.

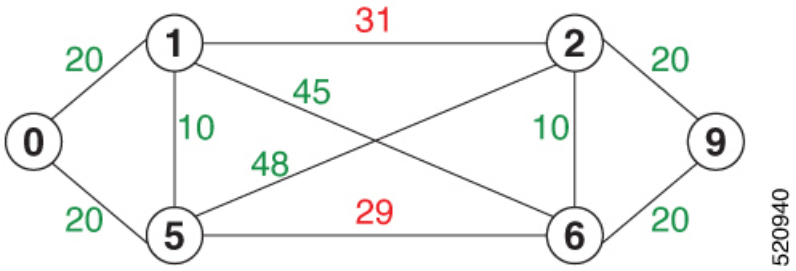
The following formula is used to calculate the normalized value:

- **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

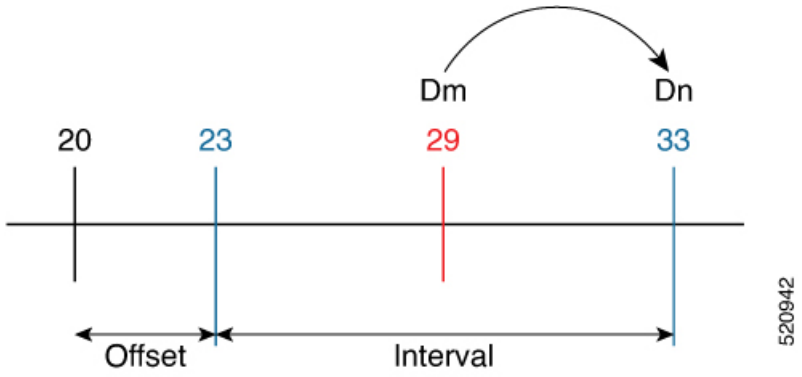
The measured delays will be normalized as follows:

- **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**

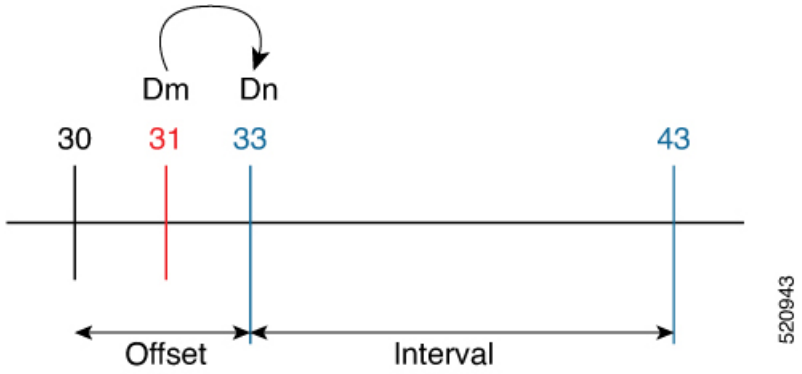


- **Dm** = 31

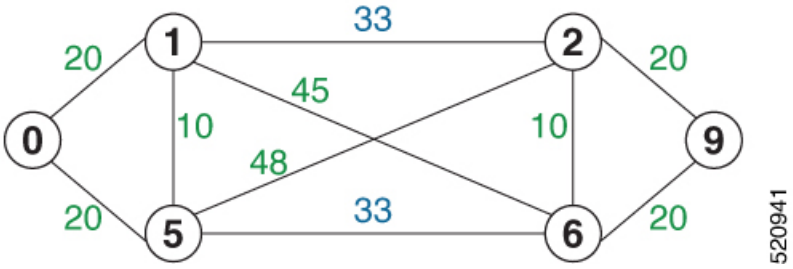
  a = 31 / 10 = 3 (3.1, rounded down to 3)

  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 11: 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 penality, 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 anomoly thresholds. The range for *upper_bound* and *lower_bound* is from 1 to 200,000 microseconds. The *lower_bound* value must be less than the *upper_bound* value.

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

Running Configuration

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

Delay Measurement for IP Endpoint

Table 12: Feature History Table

Feature Name	Release Information	Feature Description
Delay Measurement for IP Endpoint over SRv6 Network	Release 24.2.11	This feature now extends support on the Cisco NCS 540 Series routers running on Cisco IOS XR7.

Feature Name	Release Information	Feature Description
Delay Measurement for IP Endpoint over SRv6 Network	Release 24.2.1	Introduced in this release on the following Cisco NCS 540 router variants running on Cisco IOS XR: • N540-ACC-SYS • N540X-ACC-SYS • N540-24Z8Q2C-SYS In Segment Routing over an IPv6 network (SRv6), you can measure packet delay from the source to a specific IP endpoint. You can use this information for troubleshooting, network maintenance, and optimizing network performance. Additionally, you can use flow labels to verify the delay of each subsequent hop path towards the IP endpoint of that path. So that, when network traffic is distributed across multiple available paths towards an IP endpoint, delay measurement tracks the delay of each of these paths towards the IP endpoint. The feature introduces these changes: **CLI:** • The **source-address ipv6** keyword is introduced in the **performance-measurement endpoint** command. • The **segment-list name** keyword is introduced in the **segment-routing traffic-eng explicit** command. • The **flow-label** keyword is introduced in the **performance-measurement delay-profile name** command. **YANG Data Model:** • `Cisco-IOS-XR-um-performance-measurement-cfg` • `Cisco-IOS-XR-perf-meas-oper.yang` (See GitHub, YANG Data Models Navigator)
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.

UnSupported Features for IP Endpoint Over SRv6 Network

- SRv6 PM is not supported on routers that use channelized ports
- The IPv6 Endpoint in vrf (dynamic uDT6) is not supported on NCS 5508 because the timestamp is not preserved among different LCs. However, it is supported when using an SRv6 segment list with the uDT case. In this scenario, use a global source address instead of the source address within the VRF.
- The IPv6 or IPv4 Endpoint in VRF is not supported for dynamic uDT on NCS 5508.

  However, it is supported when using an SRv6 segment list with the uDT case. In this scenario, use a global source address instead of the source address within the VRF.

  ```
  For example:
  performance-measurement
    endpoint ipv4 10.2.2.2 vrf srv6_underlay
       delay-measurement
  ```

- SRv6 PM IPv6 endpoint delay session over SRv6 does not work on the Route Processor (RP) of NCS 5500 line cards and NCS 5700 line cards in the NCS 5508 Modular Router.

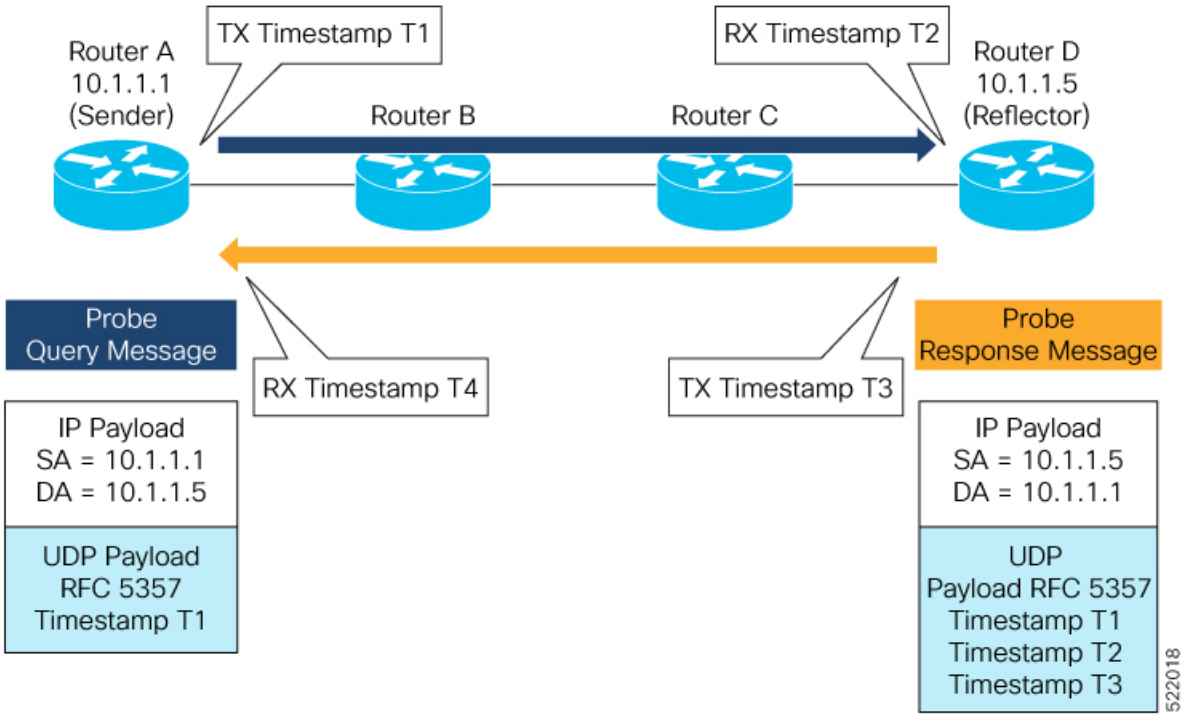
IP Endpoint Delay Measurement over MPLS Network Usecases

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 6: Delay Measurement Probe Toward an IP Endpoint Reachable in the Global Routing Table



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

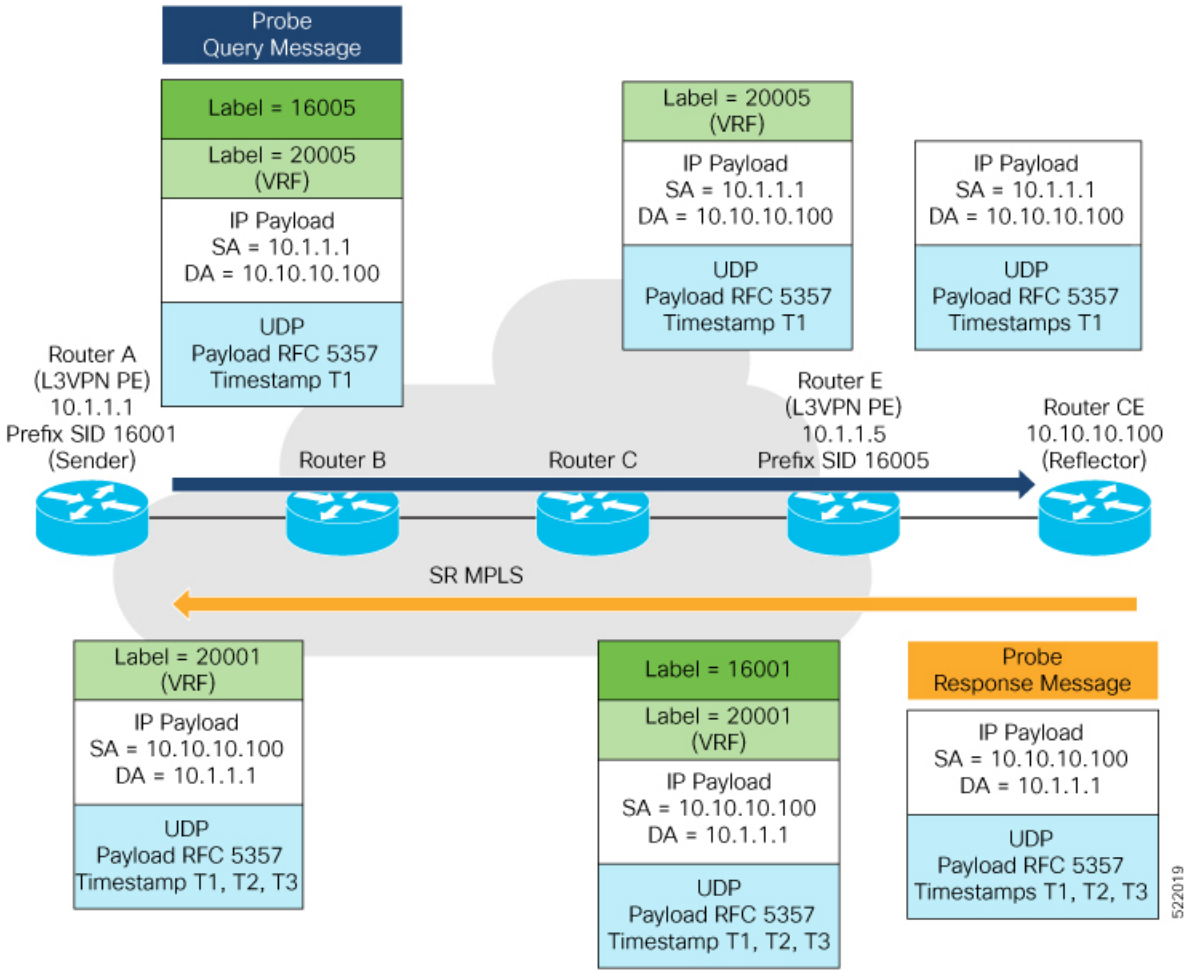
```
RouterA# show performance-measurement endpoint ipv4 10.1.1.5
```

```
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 occured

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

    Current computation:
      Started at: Jul 19 2021 16:28:06.723 (17.788 seconds ago)
      Packets  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 7: 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
```

```
  !
 !
!
```

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 occured

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

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

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

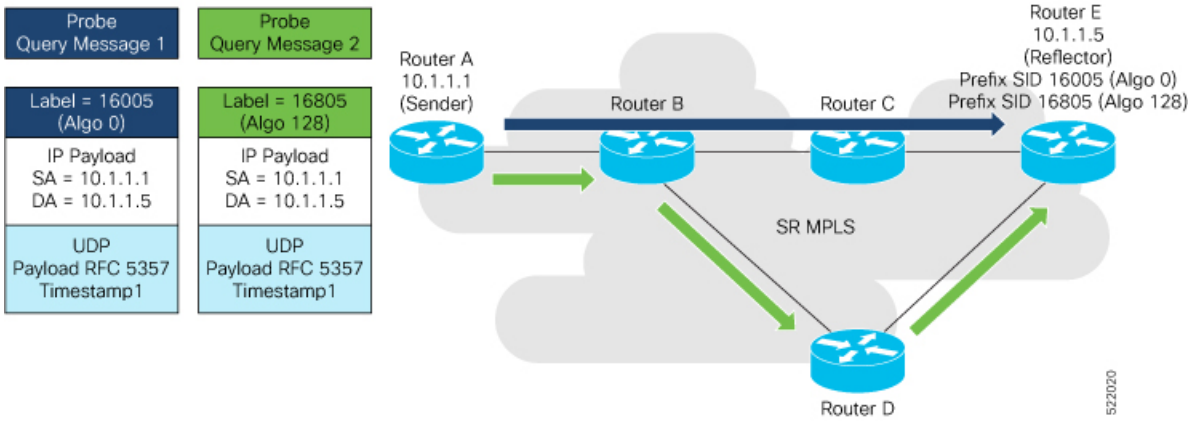
The following figure illustrates a delay measurement probe toward an IP endpoint learned by the IGP. The network interconnecting the sender and reflector provides MPLS connectivity with Segment Routing.

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

Note Starting from Cisco IOS XR Release 24.1.1, use the **segment-routing traffic-eng explicit segment-list name** command to configure the segment list.

Earlier, the **segment-routing traffic-eng segment-list name** command was used to configure the segment list.

Figure 8: 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 occured

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

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

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

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

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


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

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

  Current computation:
    Started at: Jul 19 2021 16:31:53.827 (15.844 seconds ago)
    Packets  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.

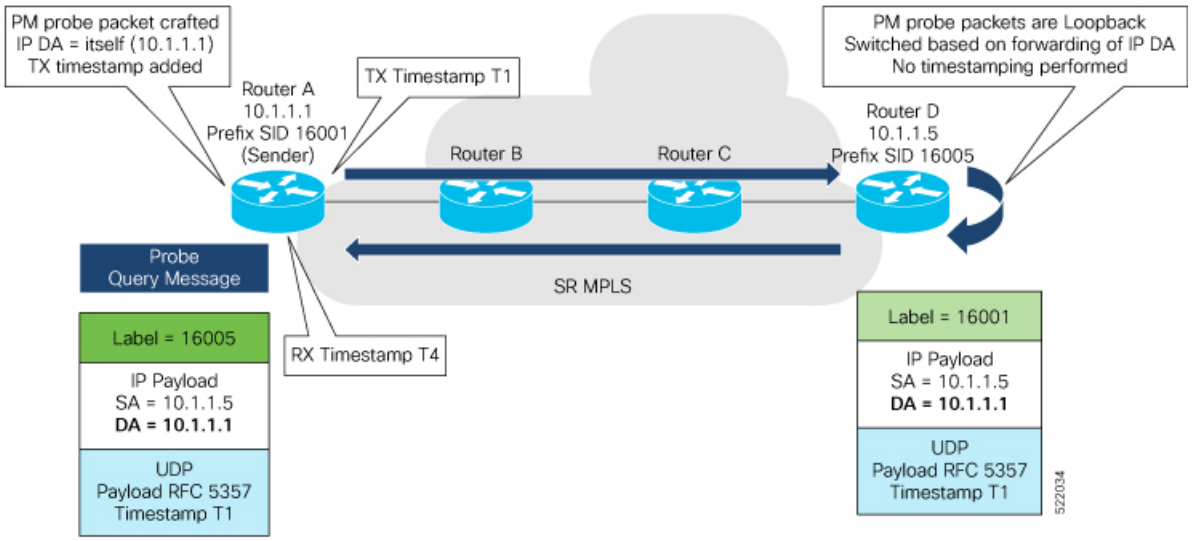
   The received timestamp (T4) stored.

   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 9: 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
RouterA(config-pm-ld-ep)# probe
RouterA(config-pm-ld-ep-probe)# measurement-mode loopback
```

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
  !
  probe
   measurement-mode loopback
  !
 !
!
end
```

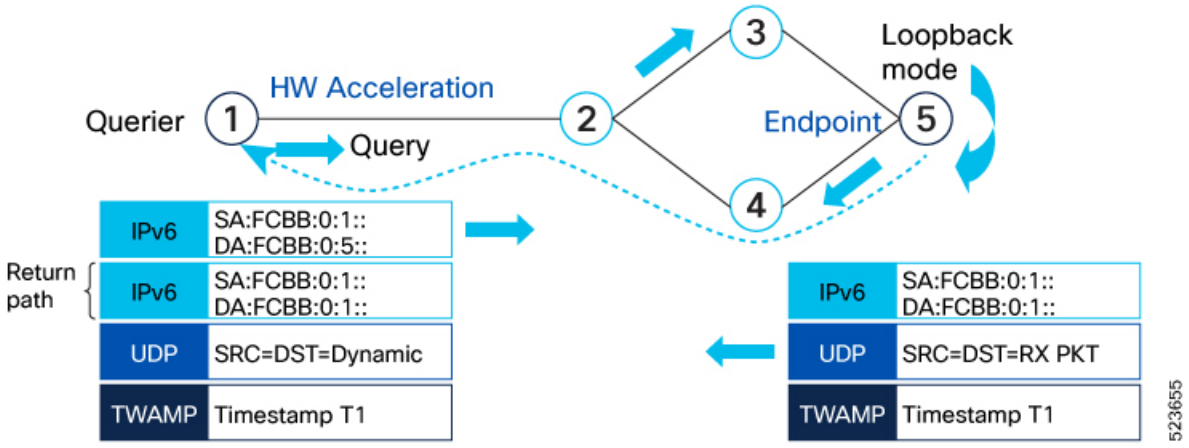
Verification

```
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
```

IP Endpoint Delay Measurement over SRv6 Network Usecase

The following figure illustrates a delay measurement probe toward an IP endpoint over the SRv6 network. The network interconnecting the sender and the reflector provides plain IP connectivity.




```
RP/0/RSP0/CPU0:ios#configure
RP/0/RSP0/CPU0:ios(config)#performance-measurement
RP/0/RSP0/CPU0:ios(config-perf-meas)#endpoint ipv6 FCBB:0:1::
RP/0/RSP0/CPU0:ios(config-pm-ep)#delay-measurement
RP/0/RSP0/CPU0:ios(config-pm-ep-dm)#delay-profile name test
RP/0/RSP0/CPU0:ios(config-pm-ep-dm)#exit
RP/0/RSP0/CPU0:ios(config-pm-ep)#exit
RP/0/RSP0/CPU0:ios(config-perf-meas)#liveness-profile name test
RP/0/RSP0/CPU0:ios(config-pm-ld-profile)#probe
RP/0/RSP0/CPU0:ios(config-pm-ld-probe)#flow-label explicit 100 200 300
RP/0/RSP0/CPU0:ios(config-pm-ld-probe)#
```

The following example shows how to use flow label for dealy profile for a default endpoint:

```
RP/0/RSP0/CPU0:ios#configure
RP/0/RSP0/CPU0:ios(config)#performance-measurement
RP/0/RSP0/CPU0:ios(config-perf-meas)#delay-profile endpoint default
RP/0/RSP0/CPU0:ios(config-pm-dm-ep)#probe
RP/0/RSP0/CPU0:ios(config-pm-dm-ep-probe)#flow-label explicit 100 200 300
```

Show Running Configuration

```
performance-measurement
 endpoint ipv6 FCBB:0:1::
  delay-measurement
```

```
   delay-profile name test
  !
 !
 liveness-profile name test
  probe
   flow-label explicit 100 200 300
  !
 !
!
```

Verification

The show output displays the delay information for the endpoint.

```
Router# show performance-measurement endpoint detail
Endpoint name: IPv6-FCBB:0:1::-vrf-default
  Source address              : 192::2
  VRF name                    : default
  Liveness Detection          : Enabled
  Profile Keys:
    Profile name              : default
    Profile type              : Endpoint Liveness Detection
  Segment-list                : None
  Liveness Detection session:
    Session ID                : 4109
    Flow-label                : 1000
    Session State: Up
    Last State Change Timestamp: Jan 23 2024 16:06:01.214
    Missed count: 0

  Liveness Detection session:
    Session ID                : 4110
    Flow-label                : 2000
    Session State: Up
    Last State Change Timestamp: Jan 23 2024 16:06:01.214
    Missed count: 0

  Segment-list                : test-dm-two-carrier-sl2
    FCBB:0:1:2:e004::/64
      Format: f3216
    FCBB:0:1:3:e000::/64
      Format: f3216
    FCBB:0:1:2:e004::/64
      Format: f3216
    FCBB:0:1:2:e000::/64
      Format: f3216
    FCBB:0:1:1:e000::/64
      Format: f3216
    FCBB:0:1:1:e004::/64
      Format: f3216
    FCBB:0:1:4:e000::/64
      Format: f3216
    FCBB:0:1:4::/48
      Format: f3216
  Liveness Detection session:
    Session ID                : 4111
    Flow-label                : 1000
    Session State: Up
    Last State Change Timestamp: Jan 23 2024 16:06:01.217
    Missed count: 0

  Liveness Detection session:
    Session ID                : 4112
    Flow-label                : 2000
```

```
Session State: Up
Last State Change Timestamp: Jan 23 2024 16:06:01.217
Missed count: 0
```

SR Policy End-to-End Delay Measurement

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 27 for more information.
- **burst interval**: Interval for sending probe packet. The default value is 3000 milliseconds and the range is from 30 to 15000 milliseconds.
- **computation interval**: Interval for metric computation. Default is 30 seconds; range is 1 to 3600 seconds.
- **protocol**:

    - **twamp-light**: SR Policy delay measurement using RFC 5357 with IP/UDP encap. This is the default protocol.

- **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
Router(config-pm-dm-srpolicy)# probe
Router(config-pm-dm-srpolicy-probe)# burst-interval 60
Router(config-pm-dm-srpolicy-probe)# computation-interval 60
Router(config-pm-dm-srpolicy-probe)# protocol twamp-light
Router(config-pm-dm-srpolicy-probe)# tos dscp 63
Router(config-pm-dm-srpolicy-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
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 occured
    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 occured
      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 occured
      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 occured
      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 occured
      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
```

Fallback delay advertisement

Table 13: Feature History Table

Feature Name	Release Information	Feature Description
Fallback delay advertisement	Release 24.4.1	You can now advertise fallback delay value, retaining delay information in performance metrics even when delay metrics for interfaces are temporarily unavailable due to hardware, synchronization, or network connectivity issues. The feature ensures optimal routing decisions by maintaining network stability and consistent performance, even when real-time metrics are temporarily unavailable. Previously, the performance metrics did not include delay metrics when they were temporarily inaccessible, resulting in visibility gaps in the network and less effective routing. The feature introduces these changes: **CLI:** **YANG Data Models:** • Cisco-IOS-XR-perfmon-measurement-cfg.yang See (GitHub, Yang Data Models Navigator)

Fallback delay advertisement for interfaces

The feature enables network operators to handle situations where GRE tunnel delay interface delay metric is not available due to hardware, synchronization, or connectivity issues. Instead of removing the interface delay metrics from the network performance metrics, which might cause network visibility loss and potential suboptimal routing decisions, network operators can configure and advertise the fallback delay value. The network can make routing decisions based on the fallback delay value instead of assuming that the link is entirely unusable.

The feature provides a backup plan for hardware failures or other issues that prevent accurate delay measurement for GRE tunnel interfaces, ensuring operational continuity by avoiding the removal of delay metrics.

Fallback delay mechanisms for seamless routing

With this feature, the network makes informed routing decisions by advertising either a maximum metric value with an anomaly flag or a user-configured fallback value.

- **Maximum Metric with an Anomaly Flag**: When you cannot measure the delay data for an interface due to hardware issues, connectivity problems, by default the performance measurement system advertises the highest possible metric value for that interface. The default fallback value advertised is 16777215 micro seconds. Also, it tags this metric with an anomaly flag, alerting the network that there is an issue with real-time delay data.
- **User-Configured Fallback Value**: When you configure a fallback value, you can advertise this fallback delay value instead of the maximum metric. The option is useful in cases where a predictable, user-defined delay value can help maintain stable routing, avoiding drastic changes in network behavior due to temporary loss of delay data.

Benefits of fallback delay advertisement for interfaces

- **Resilience**: Ensures that the network continues to make optimal routing decisions when delay metrics are not available due to hardware failures or transient session errors.
- **Predictability**: By configuring a fallback value, network administrators can ensure that the network behaves predictably in the absence of real-time delay metrics.
- **Continuity**: Prevents the removal of delay metrics, maintaining continuous visibility and routing capability.

Configure interface fallback delay value

Perform the following steps to configure a specific fallback delay value for interfaces:

Procedure

Step 1 Enable SR-PM.

Example:

```
Router#config
Router(config)#performance-measurement
```

Step 2 Set the advertise-delay fallback value for the interface to ensure that the router advertises this value when the computed delay metric is unavailable.

In the following example, the advertised interface fallback delay value is **1000**.

Example:

```
Router(config-perf-meas)#interface GigabitEthernet 0/2/0/0
Router(config-pm-intf)#delay-measurement
Router(config-pm-intf-dm)#advertise-delay fallback 1000
Router(config-pm-intf-dm)#commit
```

Step 3 Run the **show running-config** command to verify the running configuration.

Example:

```
!
performance-measurement
 interface GigabitEthernet0/2/0/0
  delay-measurement
   advertise-delay fallback 1000
  !
 !
!
```

Step 4 Run the **show performance-measurement sessions** command to verify whether the advertised metric is the default maximum metric value with an anomaly flag or the configured fallback delay value.

In the following show output, when the performance measurement session begins, by default the interface advertises the maximum metric value and the delay A flag is set.

Example:

```
Router#show performance-measurement sessions
Transport type              : Interface
Measurement type            : Delay Measurement
Interface name              : GigabitEthernet0/2/0/0
Nexthop                     : Unknown
Delay Measurement session:
  Session ID          : 4097
  Timestamp source    : Hardware (local)
  Timestamp format    : PTP
  Profile Keys:
    Profile name        : default
    Profile type        : Interface Delay Measurement
  Last advertisement:
    Advertised at: May 28 2024 13:53:41.117 (51.85 seconds ago)
    Advertised reason: Interface initial advertisement
    Advertised delays (uSec): avg: 16777215, min: 16777215, max: 16777215, variance: 0
    Packets Sent: 0, received: 0
    Min-Max A flag set: True
    Anomaly-Loss A flag set: False

  Next advertisement:
    Check scheduled at the end of the current computation (roughly every 60 seconds)
    Aggregated delays (uSec): avg: 1287, min: 623, max: 21048, variance: 628
    Packets Sent: 50, received: 50
    Rolling average (uSec): 991

    Current computation:
    Started at: May 28 2024 13:54:30.581 (1.621 seconds ago)
    Packets Sent: 2, received: 2
    Measured delays (uSec): avg: 704, min: 683, max: 724, variance: 21
    Next probe scheduled at: May 28 2024 13:54:40.570 (in 8.368 seconds)
    Next packet will be sent in 0.368 seconds
    Packet sent every 1.0 seconds
    Responder IP              : 10.10.10.2
    Number of Hops            : 1
```

Two-Way Active Measurement Protocol Light Source Address Filtering

Table 14: Feature History Table

Feature Name	Release Information	Feature Description
Two-Way Active Measurement Protocol Light Source Address Filtering	Release 7.11.1	You can now restrict unauthorized users from sending packets to the network and prevent compromising the network security and reliability. For a destination UDP port, you can configure the list of IP addresses that can send Two-Way Active Measurement Protocol (TWAMP)-light packets to responder or querier nodes. In earlier releases, the responder or querier node accepted TWAMP-light packets from all IP addresses. The feature introduces these changes: **CLI:** • The **querier** and **responder** keywords are introduced in the **performance-measurement protocol twamp-light measurement delay** command. **YANG Data Models:** • Cisco-IOS-XR-perfmeas-measurement-cfg.yang • Cisco-IOS-XR-perf-meas-oper.yang See (GitHub, Yang Data Models Navigator)

Earlier, the responder node scanned all IP addresses of a querier on the destination UDP port. In other words, the responder node accepted packets from any IP address. See Measurement Modes for more information about querier and responder nodes.

Note The responder is also called the reflector, and the querier is also called the sender.

With this configuration, you can specify the source IP addresses on both the responder and querier nodes. The responder or querier nodes accept packets only from the IP addresses configured in the TWAMP-light protocol, and reject the packets from an IP address that isn't included in the configured list.

All the configured addresses are available for use on all interfaces in the Local Packet Transport Services (LPTS). The configured address filter applies to both default and nondefault VRFs. The TWAMP delay measurement sessions use the configured addresses.

Usage Guidelines and Limitations

The following usage guidelines and limitations apply:

- When you configure the prefix entries on the responder or querier nodes, the PM adds the responder or querier node source IP address to the LPTS. For each prefix, a new LPTS entry is added or created.
- For TWAMP liveness sessions, the PM automatically adds the source IP addresses to the LPTS for if you have configured the prefix entries on the responder or querier nodes.
- As the maximum number of LPTS hardware entries are limited, ensure that enough LPTS entries are allocated for the IP addresses on a line card. You can scale the LPTS configuration to maximum LPTS entries for the PM flow-type. For more details on configuring the LTPS entries for PM flow-type, refer to the *IP Addresses and Services Configuration Guide*.

Note The PM UDP port accepts all incoming IPv4 or IPv6 packets when there are no IPv4 or IPv6 prefix entries configured.

Configure IP address on querier and responder nodes

- The length of the IPv4 and IPv6 prefixes must be less than 32 and 128 respectively.
- The length or mask of the source IP address must be:
  - For IPv4: 0–31
  - For IPv6: 0–127

Configure the IP address on a responder

Perform this task to configure the IP address of a querier on a responder node for delay measurement.

```
Router#configure
Router(config)#performance-measurement
Router(config-perf-meas)#protocol twamp-light
Router(config-pm-protocol)#measurement delay
Router(config-pm-proto-meas)#responder
Router(config-pm-proto-responder)#allow-querier
Router(config-pm-allowed-querier)#address ipv4 10.10.10.1
```

Running Configuration

```
performance-measurement
 protocol twamp-light
```

```
  measurement delay
   responder
    allow-querier
     address ipv4 10.10.10.1
    !
   !
  !
 !
!
End
```

Verification

The following example shows output from the IP address of a querier, which is configured on a responder node for delay measurement.

```
Router#show performance-measurement allowed-querier summary
Wed Oct 11 10:41:43.268 UTC

------------------------------------------------------------------------------------------------------------------------
0/RP0/CPU0
------------------------------------------------------------------------------------------------------------------------
Allowed-querier IPv4 prefix                                  : 1
  10.10.10.1/32
Allowed-querier IPv6 prefix                                  : 0

RX UDP port status:
  TWAMP-Light Default Unauthenticated responder port         : 862
    Opened IPv4 port                                         : 862
    IPv4 Port Update Time                                    : Oct 11 2023 10:37:48.118
    Opened IPv6 port                                         : 862
    IPv6 Port Update Time                                    : Oct 11 2023 10:37:47.778
```

Configure the source IP address on a querier

Perform this task to configure the IP address of a responder on a querier node for delay measurement.

```
Router#configure
Router(config)#performance-measurement
Router(config-perf-meas)#protocol twamp-light
Router(config-pm-protocol)#measurement delay
Router(config-pm-proto-meas)#querier
Router(config-pm-proto-querier)#allow-responder
Router(config-pm-allowed-responder)#address ipv4 10.10.10.1
```

Running Configuration

```
performance-measurement
 protocol twamp-light
  measurement delay
   querier
    allow-responder
     address ipv4 10.10.10.1
    !
   !
  !
 !
!
End
```

Verification

The following example shows output from the IP address of a responder, which is configured on a querier node for delay measurement.

```
Router#show performance-measurement allowed-responder summary
Wed Mar 29 19:38:06.381 UTC
------------------------------------------------------------------------------------------------------------------------

0/RP1/CPU0
------------------------------------------------------------------------------------------------------------------------


Allowed-responder IPv4 prefix                                 : 2
  10.10.10.1/32 [Auto]
  3.3.3.3/32
Allowed-responder IPv6 prefix                                 : 1
  fc00:0:1::1/128 [Auto] [Pending Add]
Querier CPU UDP port status:
  TWAMP-Light Default Unauthenticated querier port            : N/A
    Opened IPv4 port                                          : 27643
    IPv4 Port Update Time                                     : Mar 29 2023 18:43:49.080
    Opened IPv6 port                                          : 28274
    IPv6 Port Update Time                                     : Mar 24 2023 20:58:46.150
```

Synthetic Loss Measurement

Table 15: Feature History Table

Feature Name	Release	Description
Synthetic Loss Measurement	Release 24.1.1	You can now proactively monitor and address potential network issues before they impact users by measuring key parameters everywhere, packet loss, and jitter. Using this information, you can plan network capacity optimally and ensure quality of service. Such proactive action is possible because this feature reports synthetic Two-Way Active Measurement Protocol (TWAMP) test packets deployed in delay-profile or delay measurement sessions. It also enables you to set the upper and lower limits and notifies when the synthetic packet loss metric is out of the set limit. The feature introduces these changes: **CLI:** • The optional **anomaly-loss** and **anomaly-check** keywords are introduced in the **performance-measurement delay-profile** command. • **show performance-measurement history** **YANG Data Model** • New XPaths for `Cisco-IOS-XR-um-performance-measurement-cfg` • New operations for `Cisco-IOS-XR-perf-meas-oper.yang` (see GitHub, YANG Data Models Navigator)

Synthetic packets are data packets that are artificially generated for the purpose of testing or managing a network, as opposed to being generated by normal network data traffic. These packets are unrelated to regular user activities or data. Instead, they are intentionally designed to serve monitoring, testing, and diagnostic purposes.

Synthetic TWAMP test packets deployed in delay profile are used to simulate network traffic, test network performance, and evaluate network infrastructure. They are particularly useful for assessing the quality of service (QoS) of a network, identifying network vulnerabilities, and troubleshooting network issues. These packets enable network administrators to identify potential problems before they impact the overall network performance and reliability. *For more information on TWAMP, refer to the IP Service Level Agreements chapter in the System Monitoring Configuration Guide for Cisco NCS 560 Series Routers*.

Overall, synthetic packets play a critical role in network management, providing network engineers with a powerful tool for optimizing network performance, identifying and resolving issues.

Note This is an inbuilt feature of delay measurement. To get the synthetic packet loss information for delay-measurement sessions, you must only configure the delay sessions. No additional configuration is required for Synthetic Loss Measurement. However, an optional **anomaly-loss** command is introduced.

Synthetic packets simulate various types of traffic that can be sent with the normal traffic to measure loss or latency, throughput or packet loss. This is often done in network performance monitoring and testing, where the goal is to understand how the network or a specific network device performs under typical conditions.

This feature enables you to measure and monitor the Synthetic packets lost and help make informed decisions on the following:

- **Network Performance Monitoring**: Synthetic packets can be used to simulate different types of network traffic in order to measure and test the performance of network devices or the network itself.
- **Troubleshooting**: They can be used to diagnose problems in a network by sending them through different parts of the system to see where they may fail or be slowed down.
- **Security**: They can also be used in security applications, such as penetration testing, where synthetic packets are sent to a system to find vulnerabilities.

Configure Synthetic Loss measurement

The following example enables Synthetic Loss Measurement for an SR Policy.

Note A delay-measurement session is good enough to get the synthetic packet loss automatically, without any extra configuration. The configure shown here is for anomaly loss, which is optional.

```
Router(config)#performance-measurement
Router(config-perf-meas)#delay-profile sr-policy default
Router(config-pm-dm-srpolicy)#advertisement
Router(config-pm-dm-srpolicy-adv)#anomaly-loss
Router(config-pm-dm-srpolicy-adv-anom-loss)#upper-bound 30 lower-bound 20
Router(config-pm-dm-srpolicy-adv-anom-loss)#commit
```

Note Similarly, you can configure Synthetic Loss Measurement for **endpoint default**, **interface default** or **name** (named profile).

Running Configuration

The running configuration for this feature is as shown:

```
performance-measurement
  delay-profile sr-policy default
    advertisement
```

```
      anomaly-loss
        upper-bound 30 lower-bound 20
        !
      !
   !
 !
!
```

Verification

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

```
Router# show performance-measurement interfaces delay detail
Interface Name: GigabitEthernet0/2/0/0 (ifh: 0x1000020)
...
      Last advertisement:
      ...
      Packets  40, received: 40
      ...
    Next advertisement:
      ...
      Packets  10, received: 10
      ...
    Current computation:
      ...
      Packets  4, received: 4
```

```
Router# show performance-measurement history probe-computation interfaces
Interface Name: GigabitEthernet0/2/0/0 (ifh: 0x1000020)
Delay-Measurement history (uSec):
  Probe Start Timestamp     Pkt(TX/RX)   Average       Min       Max
  Aug 01 2023 08:04:15.230       10/10       704       651       779
```

Endpoint

Use these show commands to verify the Endpoint configuration. The example include endpoint delay details and advertisement history. You can also verify the aggregation history.

```
Router# show performance-measurement endpoint delay detail
Endpoint name: IPv4-192.168.0.4-vrf-default
 ...
    Last advertisement:
      ...
       Packets  40, received: 40
      ...
    Next advertisement:
     ...
      Packets  30, received: 30
     ...
    Current computation:
     ...
     Packets  6, received: 6
     ...
```

```
Router# show performance-measurement history advertisement endpoint
Endpoint name: IPv4-192.168.0.4-vrf-default
  ...
    Delay-Measurement history (uSec):
```

```
      Advertisement Timestamp   Pkt(TX/RX)   Average       Min       Max        Reason
      Aug 01 2023 08:31:18.835       40/40      3948      3127     15503       PER-MAX
```

RSVP-TE

Use these show commands to verify the RSVP-TE configuration. The example include rsvp-te delay details and the aggregation history. You can also verify the advertisement history.

```
Router# show performance-measurement rsvp-te detail
Tunnel name: tunnel-te1 (ifh: 0xd0)
  ...
    Last advertisement:
      ...
      Packets  40, received: 40
      ...
    Next advertisement:
      ...
      Packets  10, received: 10
      ...
    Last computation:
      ...
      Packets  10, received: 10
      ...
    Current computation:
      ...
      Packets  6, received: 6
      ...
  ...
```

```
Router# show performance-measurement history aggregation rsvp-te
...
Delay-Measurement history (uSec):
      Aggregation Timestamp     Pkt(TX/RX)   Average       Min       Max
      Aug 01 2023 08:37:23.702       40/40      3372      3172      4109
...
```

SR-Policy

Use these show commands to verify the SR-Policy configuration. The example include delay details and advertisement history. You can also verify the aggregation history.

```
Router# show performance-measurement sr-policy delay detail
SR Policy name: srte_c_10_ep_192.168.0.4
...
    Last advertisement:
      ...
      Packets  88, received: 88
      ...
    Next advertisement:
      ...
      Packets  132, received: 132
      ...
    Last computation:
      ...
      Packets  4, received: 4
      ...
    Current computation:
      ...
      Packets  4, received: 4
      ...
  Segment-List              : R4
    ...
    Last advertisement:
```

```
      ...
      Packets  88, received: 88
      ...
    Next advertisement:
      ...
      Packets  132, received: 132
      ...
    Last computation:
      Packets  4, received: 4
      ...
    Current computation:
      Packets  4, received: 4
      ...
```

```
Router# show performance-measurement history advertisement sr-policy
...
Delay-Measurement history (uSec):
          Advertisement Timestamp   Pkt(TX/RX)   Average       Min       Max        Reason
          Aug 01 2023 10:05:14.072       24/24      3408      3408      3408     ACCEL-MAX
```

Packet loss advertisement

Use this show command to verify multiple configurations with a single show command:

```
Router# show performance-measurement history advertisement sr-policy/endpoint
...
Delay-Measurement history (uSec):
          Advertisement Timestamp   Pkt(TX/RX)   Average       Min       Max        Reason
          Sep 14 2023 11:35:38.180       10/9    3595        3411      3696     NEW-SESSION

          Sep 14 2023 11:34:38.178       10/0     0           0         0      SESSION-ERROR

          Sep 14 2023 11:34:08.177       10/7    3733        3733      3733     ANOM-PKT-LOSS

           Sep 14 2023 11:34:02.177         8/7    3733        3733      3733      PKT-LOSS
           Sep 14 2023 11:33:38.176       10/10    3823        3617      4627        FIRST
...
```

```
Router# show performance-measurement sessions
...
Last advertisement:
    Advertised at: Sep 14 2023 08:47:16.540 (145.777 seconds ago)
    Advertised reason: Periodic timer, max delay threshold crossed
    Advertised delays (uSec): avg: 5373, min: 3992, max: 26212, variance: 0
    Packets  30, received: 30
    Min-Max A flag set: False
    Loss A flag set: False
...
...
Last advertisement:
    Advertised at: Sep 14 2023 08:52:39.603 (12.522 seconds ago)
    Advertised reason: PM session anomaly due to packet loss
    Advertised delays (uSec): avg: 5373, min: 3992, max: 26212, variance: 0
    Packets  30, received: 30
    Min-Max A flag set: False
    Loss A flag set: True
...
Last advertisement:
    Advertised at: Sep 15 2023 11:42:41.594 (47.660 seconds ago)
    Advertised reason: Performance measurement session error
...
```

Note No metrics or A bit is shown when there is a session error.

```
Router# show logging
RP/0/0/CPU0:Sep 14 11:33:38.176 PDT: perf_meas[1001]:
%ROUTING-PERF_MEAS-5-PM_DELAY_EXCEEDS_THRESHOLD : Reason: First advertisement, PM delay
metric avg: 3823, min: 3617, max: 4627, var: 138, min-max anomaly flag: Not set, pkt-loss
anomaly flag: Not set, exceeded threshold on SR Policy N:srte_c_10_ep_192.168.0.4, L:2
RP/0/0/CPU0:Sep 14 11:34:08.177 PDT: perf_meas[1001]:
%ROUTING-PERF_MEAS-5-PM_DELAY_EXCEEDS_THRESHOLD : Reason: PM session anomaly due to packet
 loss, PM delay metric avg: 0, min: 0, max: 0, var: 0, min-max anomaly flag: Not set,
pkt-loss anomaly flag: Set, exceeded threshold on SR Policy N:srte_c_10_ep_192.168.0.4, L:2

RP/0/0/CPU0:Sep 14 11:35:38.180 PDT: perf_meas[1001]:
%ROUTING-PERF_MEAS-5-PM_DELAY_EXCEEDS_THRESHOLD : Reason: PM  session anomaly  due to packet
 loss, PM delay metric avg: 3595, min: 3411, max: 3696, var: 58, min-max anomaly flag: Not
 set, pkt-loss anomaly flag: Not set, exceeded threshold on SR Policy
N:srte_c_10_ep_192.168.0.4, L:2

============================================================================================
```

In this scenario, the router was running normally and then advertised a packet loss anomaly, the flag is set; later the router resumed to normal, the flag is unset, and it advertised a packet loss anomaly again.

Delay and synthetic loss measurement for GRE tunnel interfaces

Table 16: Feature History Table

Feature Name	Release Information	Feature Description
Delay and synthetic loss measurement for GRE tunnel interfaces	Release 24.4.1	You can now measure the latency or delay experienced by data packets when they traverse a network, and also proactively monitor and address potential network issues before they impact users by measuring key parameters such as packet loss, and jitter for GRE tunnel interfaces. This feature enables you to report synthetic Two-Way Active Measurement Protocol (TWAMP) test packets that are deployed in delay-profile or delay measurement sessions, and enables delay measurement for GRE tunnel interfaces. The feature introduces these changes: **CLI:** The **performance-measurement interface** command supports the **tunnel-ip** keyword.

Delay and synthetic loss measurement for GRE tunnel interfaces

With this feature, you can configure a GRE interface with delay measurement under PM. You can now run sessions over a GRE tunnel as an interface session, and the PM advertises these measurement metrics to the Interface Manager (IM).

The PM for delay measurement uses the IP/UDP packet format that is defined in RFC 5357 (TWAMP-Light) for probes.

Synthetic packets are data packets that are artificially generated for testing or managing a network, as opposed to being generated by normal network data traffic. These packets are unrelated to regular user activities or data. Instead, they are intentionally designed to serve monitoring, testing, and diagnostic purposes. With synthetic loss measurement, you can set the upper and lower limits and notify when the synthetic packet loss metric is out of the set limit.

For more information about delay and synthetic loss measurement, see the Delay Measurement, on page 27 and Synthetic Loss Measurement, on page 75 sections.

Measurement modes for GRE tunnel interfaces

GRE tunnel interfaces support these modes to measure delay and synthetic loss:

- One-way
- Two-way

Both one-way and two-way measurement modes support round-trip synthetic loss measurement.

Usage guidelines and limitations for delay and synthetic loss measurement for GRE tunnels

Configure delay and synthetic loss measurement for GRE tunnel interfaces

Procedure

Step 1 Run the **interface tunnel-ip** command to configure the tunnel mode, source, and destination for the interface.

Note
You must configure the responder node to decapsulate the GRE tunnel and establish a proper return path to the querier node.

In the following example, the tunnel uses GRE in IPv4 mode, encapsulating or decapsulating IP packets for transport across the tunnel. The source IP address of the tunnel is set to 209.165.201.2, indicating the origin of the IP packets. The destination IP address is set to 209.165.201.3, indicating where the encapsulated packets are sent.

Example:

```
Router#config
Router(config)#interface tunnel-ip 23
Router(config-if)#ipv4 address 209.165.201.1/27
Router(config-if)#ipv6 address 2001:DB8::/32
Router(config-if)#tunnel mode gre ipv4 encap
Router(config-if)#tunnel mode gre ipv4 decap
Router(config-if)#tunnel source 209.165.201.2
Router(config-if)#tunnel destination 209.165.201.3
Router(config-pm-intf-dm)#commit
```

Step 2 Enable delay measurement on the configured GRE tunnel interface.

Example:

```
Router(config)#performance-measurement
Router(config-perf-meas)#interface tunnel-ip 23
Router(config-pm-intf)#delay-measurement
Router(config-pm-intf-dm)#commit
```

Step 3 Run the **show running-config** command to verify the running configuration.

Example:

```
!
interface tunnel-ip23
```

```
 ipv4 address 209.165.201.1 255.255.255.224
 ipv6 address 2001:db8::/32
 tunnel mode gre ipv4 decap
 tunnel source 209.165.201.2
 tunnel destination 209.165.201.3
!
performance-measurement
 interface tunnel-ip23
  delay-measurement
  !
 !
!
```

Step 4 Run the **performance-measurement delay-profile** command to configure synthetic loss measurement for the GRE tunnel interface.

For more information about configuring synthetic loss measurement, see the Configure Synthetic Loss measurement, on page 76 section.

Step 5 Run the **show performance-measurement interfaces delay** command to verify the delay and synthetic loss measurement that is enabled for the GRE tunnel interface.

Example:

```
Router#show performance-measurement interfaces delay detail
Mon Nov  4 12:44:56.849 UTC

------------------------------------------------------------------------------------------------------------------------
0/RP0/CPU0
------------------------------------------------------------------------------------------------------------------------

Interface Name: tunnel-ip23 (ifh: 0x78000014)
  Delay-Measurement                 : Enabled
  Loss-Measurement                  : Enabled
  Path-Tracing                      : Disabled
  Configured IPv4 Address           : 209.165.201.1
  Configured IPv6 Address           : 2001:db8::
  Link Local IPv6 Address           : fe80::657e:adff:feec:d876
  Configured Next-hop Address       : Unknown
  Local MAC Address                 : 0000.0000.0000
  Next-hop MAC Address              : 0000.0000.0000
  In-use Source Address             : Unknown
  In-use Destination Address        : Unknown
  Primary VLAN Tag                  : None
  Secondary VLAN Tag                : None
  State                             : Up

  Delay Measurement session:
    Session ID          : 4097
    Timestamp source    : Hardware (local)
    Timestamp format    : PTP
    Profile Keys:
      Profile name      : default
      Profile type      : Interface Delay Measurement
    Last advertisement:
      Advertised at: Nov 04 2024 11:26:41.674 (4695.200 seconds ago)
      Advertised delays (uSec): avg: 0, min: 0, max: 0, variance: 0
      Packets sent: 40, received: 40
      Min-Max A flag set: False
      Anomaly-Loss A flag set: False

    Next advertisement:
      Threshold check scheduled in 1 more probe (roughly every 120 seconds)
      Aggregated delays (uSec): avg: 2265, min: 1967, max: 4572, variance: 248
```

```
    Packets Sent: 20, received: 20
    Rolling average (uSec): 2219

  Current computation:
    Started at: Nov 04 2024 11:31:59.071 (14.416 seconds ago)
    Packets Sent: 5, received: 5
    Measured delays (uSec): avg: 2405, min: 2232, max: 2714, variance: 173
    Next probe scheduled at: Nov 04 2024 11:32:29.053 (in 15.566 seconds)
    Next packet will be sent in 0.566 seconds
    Packet sent every 3.0 seconds
    Responder IP            : 192::1
    Number of Hops          : 1
```

Delay Measurement Using Software Timestamp

Table 17: Feature History Table

Feature Name	Release Information	Feature Description
Delay Measurement Using Software Timestamp	Release 24.3.1	You can now identify performance issues caused by the network, disk I/O, processing, or other factors using software timestamping on your router by measuring the delay and loss of each network path, even if the existing hardware lacks timestamp support. The feature introduces these changes: **CLI:** • The **timestamp-format NTP** keyword is introduced in the **performance-measurement delay-profile** command.

Timestamps for Delay Measurement

The Performance Measurement (PM) is a crucial mechanism that measures the delay and loss on network paths. It measures the delay and loss by sending and receiving timestamped probe packets at the ingress and egress nodes.

PM uses probe packets to carry specific Type-Length-Value (TLVs) to encode the measurement information for delay measurement. One of the TLVs used for delay measurement is the Timestamp TLV that contains the time when a node sends or receives the probe packet.

The Timestamp TLV can have different formats, such as Precision Time Protocol (PTP) or Network Time Protocol (NTP).

You need to insert the Timestamp TLV close to the physical interface where the packet is transmitted or received to accurately measure the delay.

You can measure delay and loss using hardware timestamp. If the hardware does not support the timestamp capability and the corresponding format and offset, software timestamp is an option that you need to consider.

Before Release 24.3.1, only hardware timestamping was supported.

Software Timestamps for Delay Measurement

A software timestamp is a method that uses software for the insertion of Type-Length-Value (TLVs) into probe packets..

In instances where the hardware device does not support timestamping, we enable the router to set the timestamp in software, which helps insert the Type-Length-Value (TLVs) into the probe packets. The default timestamp format is PTP. You can configure to use the NTP format using the **timestamp-format NTP** command.

Measurement Modes for Software Timestamp

The software timestamp supports these three modes:

- One-way measurement mode
- Two-way measurement mode
- Loopback measurement mode

For more information, see Measurement Modes, on page 27.

Supported Features for Software Timestamp

The software timestamp supports these performance measurement features:

- Link delay measurement
- SR Policy
- IP Endpoint with both MPLS and SRv6
- RSVP-TE LSPs

Key Concepts

- **Delay Measuremnt** : Delay measuremnt is a mechanism used to measure the latency or delay experienced by data packets when they traverse a network. For more information, see Delay Measurement.
- **NTP** : Network Time Protocol (NTP) provides time synchronization to all devices on a network. The primary NTP servers are synchronized to a reference clock, such as GPS receivers and telephone modem services.

  For more information, see *System Management Configuration Guide for Cisco NCS 560 Series Routers*
- **PTP**: Precision Time Protocol (PTP) defines a method for distributing time across a network with its foundation based on the IEEE 1588-2008 standard.

  For more information, see *System Management Configuration Guide for Cisco NCS 560 Series Routers*

Benefits of Software Timestamp

Software Timestamp helps you

- provide a software-based timestamping fallback option when the hardware lacks support for the necessary timestamping capability, format, or offset.
- insert the timestamp to probe packets when a hardware device cannot support the required format or offset.
- enable the packet loss measurement automatically since the software can also insert the Counter TLV, which contains the sequence number of the probe packet.

How Delay is Measured Using Software Timestamp?

Here is how you measure a delay using software timestamp:

1. Event Initiation: When a client sends a packet, the current time is recorded as a timestamp using PTP or NTP.
2. Event Completion: When the client recieves an acknowledgment from the server, another timestamp is recorded.
3. Delay Calculation: The delay is calculated by subtracting the first event's timestamp from the second event's timestamp. This calculation provides the round-trip time (RTT) or one-way delay, depending on the specific context.
4. Accuracy and Precision: The accuracy of software timestamps relies on the resolution of the system clock and any potential delays introduced by the software layers handling the timestamping.

Guidelines and Limitations for Software Timestamp

Here are the some of the guidelines and limitations that you need to keep in mind before you configure software timestamp:

- For devices that support hardware timestamping, the hardware timestamp has priority over the software timestamp.
- One-way delay measurement is not supported when there is a mix of hardware and software timestamp ping.
- Delay Measurement using Software Timestamp is not supported on the Compatibility mode of NCS 5500 modular routers using NCS 5500 line cards and NCS 5700 line cards.

Configure Software Timestamp for Delay Measurement

The default software timestamp format is PTP. You can enable the router to use the NTP format using the **timestamp-format NTP** command.

You can configure software timestamp for

- Link delay

- SR Policy
- IP Endpoint with both MPLS and SRv6, and
- RSVP-TE LSPs.

Procedure

Step 1 Use the **timestamp-format NTP** command to configure software timestamp with NTP format on a router.

- This example shows how to configure a software timestamp for IP Endpoint.

```
Router#configure
Router(config)# performance-measurement
Router(config-perf-meas)#delay-profile endpoint default
Router(config-pm-dm-ep)#probe
Router(config-pm-dm-ep-probe)#timestamp-format NTP
Router(config-pm-dm-ep-probe)#
```

- This example shows how to configure software timestamp for SR policy.

```
Router#configure
Router(config)# performance-measurement
Router(config-perf-meas)#delay-profile sr-policy default
Router(config-pm-dm-srpolicy)#probe
Router(config-pm-dm-srpolicy-probe)#timestamp-format NTP
```

- This example shows how to configure software timestamp for delay profile name.

```
Router#configure
Router(config)# performance-measurement
Router(config-perf-meas)#delay-profile name blue
Router(config-pm-dm-profile)#probe
Router(config-pm-dm-probe)#timestamp-format NTP
Router(config-pm-dm-probe)#
```

- This example shows how to configure software timestamp for interface delay profile.

```
Router#configure
Router(config)# performance-measurement
Router(config-perf-meas)#delay-profile interfaces default
Router(config-pm-dm-intf)#probe
Router(config-pm-dm-intf-probe)#timestamp-format NTP
Router(config-pm-dm-intf-probe)#
```

- This example shows how to configure software timestamp for RSVP-TE LSPs.

```
Router#configure
Router(config)# performance-measurement
Router(config-perf-meas)#delay-profile rsvp-te default
Router(config-pm-dm-rsvpte)#probe
Router(config-pm-dm-rsvpte-probe)#timestamp-format NTP
Router(config-pm-dm-rsvpte-probe)#
```

Step 2 Execute the **show performance-measurement sessions** command to verify the timestamp format.

- This example shows how to verify the timestamp format for the IP endpoint:

```
Router# show performance-measurement endpoint detail
Endpoint name: IPv4-192.168.0.4-vrf-default
  Source address              : 192.168.0.2
```

```
  Destination Port            : 862
  Source Port                 : 35923
  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          : 4275
    Timestamp source    : Software (local)
    Timestamp format    : NTP
    Last advertisement:
      No advertisements have occured

    Next advertisement:
      Threshold check scheduled in 1 more probe (roughly every 30 seconds)
      No probes completed

    Current computation:
      Started at: Apr 01 2024 17:31:41.452 (11.141 seconds ago)
      Packets Sent: 5, received: 5
      Measured delays (uSec): avg: 3538, min: 2836, max: 4379, variance: 702
      Next probe scheduled at: Apr 01 2024 17:32:08.459 (in 15.866 seconds)
      Next packet will be sent in 0.866 seconds
      Packet sent every 3.0 seconds
      Responder IP            : 192.168.0.4
      Number of Hops          : 2
      Probe samples:
        Packet Rx Timestamp       Measured Delay (nsec)
        Apr 01 2024 17:31:41.457         3507559
        Apr 01 2024 17:31:41.480         2836021
        Apr 01 2024 17:31:44.481         3072131
        Apr 01 2024 17:31:47.483         3896705
        Apr 01 2024 17:31:50.484         4379536
-------------------------------------
.
.
.
```

- This example shows how to verify the timestamp format for the SR policy:

```
Router# show performance-measurement sessions
------------------------------------------------------------
Transport type              : SR Policy
Measurement type            : Delay Measurement
Policy name                 : srte_c_10_ep_192.168.0.4
Color                       : 10
Endpoint                    : 192.168.0.4
Instance                    : 2
preference                  : 100
Protocol-origin             : Configured
Discriminator               : 100
Segment-list                : R4
Atomic path:
  Hops                      : 192.168.0.4
  Session ID                : 4145
  Trace ID                  : 2101013379
  NPU Offloaded session     : False
  Timestamping Enabled      : True
Timestamp source          : Software (local)
  Timestamp format          : NTP
```

```
  Last advertisement:
    No advertisements have occured
  Next advertisement:
    No probes completed
  Last computation:
    None
  Current computation:
    Packets Sent: 2, received: 2
    Measured delays (uSec): avg: 3531, min: 3424, max: 3639, variance: 107
  Probe samples:
    Packet Rx Timestamp       Measured Delay (nsec)
    Apr 01 2024 16:47:26.876         3424443
    Apr 01 2024 16:47:27.927         3639431
  Liveness Detection: Disabled
  Responder IP              : 192.168.0.4
  Number of Hops            : 2
```

- This example shows how to verify the timestamp format for interfaces:

```
Router# show performance-measurement profile default interface delay
Interface Delay Measurement (default)
  Profile configuration:
    Measurement mode : Two-way
    Timestamp format : NTP
    Protocol type: TWAMP-light
    Encap mode  : UDP
    Type of service:
      PM-MPLS traffic class   : 6
      TWAMP-light DSCP  : 48
    Probe computation interval : 30 (effective: 30) seconds
    Burst interval : 3000 (effective: 3000) mSec
    TX interval  : 3000000 (effective: 3000000) uSec
    Packets per computation interval : 10
    Periodic advertisement : Enabled
      Interval : 120 (effective: 120) sec
      Threshold : 10%
      Minimum-change : 500 uSec
    Advertisement accelerated : Disabled
    Anomaly delay advertisement: Disabled
    Anomaly loss advertisement: Disabled
    Advertisement logging:
      Delay exceeded : Disabled (default)
    Threshold crossing check : Minimum-delay
    Router alert  : Disabled (default)
    Destination sweeping mode : Disabled
    Flow Label Count: 0
```

- This example shows how to verify the timestamp format for RSVP TE:

```
Router#show performance-measurement profile default rsvp-te delay
RSVP-TE Delay Measurement (default)
  Profile configuration:
    Measurement mode : One-way
    Timestamp format : NTP
    Protocol type   : TWAMP-light
    Encap mode  : UDP
    Type of service:
      PM-MPLS traffic class : 6
      TWAMP-light DSCP : 48
    Probe computation interval   : 30 (effective: 30) seconds
    Burst interval  : 3000 (effective: 3000) mSec
    TX interval    : 3000000 (effective: 3000000) uSec
    Packets per computation interval  : 10
    Periodic advertisement : Enabled
```

```
  Interval   : 120 (effective: 120) sec
  Threshold  : 10%
  Minimum-change  : 500 uSec
Advertisement accelerated : Disabled
Anomaly delay advertisement  : Disabled
Anomaly loss advertisement : Disabled
Advertisement logging:
  Delay exceeded  : Disabled (default)
Threshold crossing check : Maximum-delay
Router alert  : Disabled (default)
Destination sweeping mode  : Disabled
Flow Label Count : 0
```