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 profiles for different types of delay measurements. Delay profile type interfaces is used for link-delay measurement. Delay profile type sr-policy is used for SR policy delay measurements. Delay profile allows you to schedule probe and configure metric advertisement parameters for delay measurement.
Protocols	The TWAMP Light from Appendix of RFC 5357 is standardized as Simple TWAMP in RFC 8762. Then it was extended with RFC 8972.
Probe and burst scheduling	Schedule probes and configure metric advertisement parameters for delay measurement.
Metric advertisements	Advertise measured metrics periodically using configured thresholds. Also supports accelerated advertisements using configured thresholds.
Measurement history and counters	Maintain packet delay and loss measurement history and also session counters and packet advertisement counters.

Usage Guidelines and Limitations

Performance Measurement (PM) probes typically follow the designated Segment Routing Traffic Engineering (SR-TE) path. However, in certain scenarios, the convergence of the PM probes and the SR-TE path may occur at different times. During this convergence period, PM probes may temporarily follow the IGP path and utilize an alternate egress interface until full convergence is achieved.

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.

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
IP Endpoint Liveness Monitoring	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8711-48Z-M routers.
IP Endpoint 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's IP address can be located in the global routing table or under a user-specified VRF routing table.

  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.

Supported and Unsupported Features

The table lists supported and unsupoted features for Liveness Monitoring for IP Endpoint over SRv6 Network.

Table 3: Supported and Unsupported Features for Liveness Monitoring for IP Endpoint over SRv6 Network

Supported Features	Unsupported Features
SRv6 Endpoint Liveness in Default VRF Example: *endpoint ipv6 fccc:2:: liveness-detection* In this example, the endpoint with the IPv6 address fccc:2:: is the SRv6 uSID format.	IPv6 Endpoint Liveness in Default VRF (over SRv6) Example: *endpoint ipv6 10::2 liveness-detection* In this example, the endpoint with the IPv6 address 2::2 is part of an underlay network using SRv6.
IPv6 Endpoint Liveness in VRF (static uDT6) Example: *endpoint ipv6 fccc:2:fe02:: liveness-detection* In this example, the endpoint with the IPv6 address fccc:2:fe02:: is the static uDT6 uSID carrier	IPv6 Endpoint Liveness in VRF (dynamic DT6 encap) Example: *endpoint ipv6 10::1 vrf purple source-address ipv6 10::2 liveness-detection*

Supported Features	Unsupported Features
IPv4 Endpoint Liveness in VRF or GRT (static uDT4) Example: *endpoint ipv4 10.5.56.1 source-address ipv4 10.1.17.1 segment-routing traffic-eng explicit segment-list name vrf-2-3-5-udt4 liveness-detection* The segment-list *vrf-2-3-5-udt4* here has static uDT4 SID.	IPv4 Endpoint Liveness in VRF or GRT (dynamic uDT4 encap Example: *endpoint ipv4 10.0.0.1 vrf purple source-address ipv4 10.0.0.2 liveness-detection*
	IPv6 address over SRV6 underlay Example: *endpoint ipv6 10::1 source-address ipv6 10::2 liveness-detection*

Usage Guidelines and Limitations

- For liveness detection, the session fails to come up when the endpoint address is a regular IPv4 address in a default VRF and that is a normal loopback IP address that uses IGP path. Packets get dropped with the following message. However, this issue does not apply if a segment list is configured.

  ```
  GRE IPv4 decap qualification failed
  ```

  To mitigate this issue, you must configure the GRE tunnel on responder. The following example shows how to configure GRE tunnel:

  ```
  /*Tunnel config on responder*\
  interface tunnel-ip1
  tunnel model ipv4 decap
  tunnel source 10.3.1.1
  tunnel destination 10.1.1.1
  ```

- Liveness session without segment list for an endpoint in a non-default VRF is not supported.
- SR Performance Measurement endpoint session over BVI interface is not supported.
- SRv6 locator prefix and VRF SRv6 locator/function (uDT4/uDT6) as IPv6 endpoint of a probe is not supported.
- IPv6 Endpoint Liveness in Default VRF is not supported over SRv6.
- PM probe over GREv4 is 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.

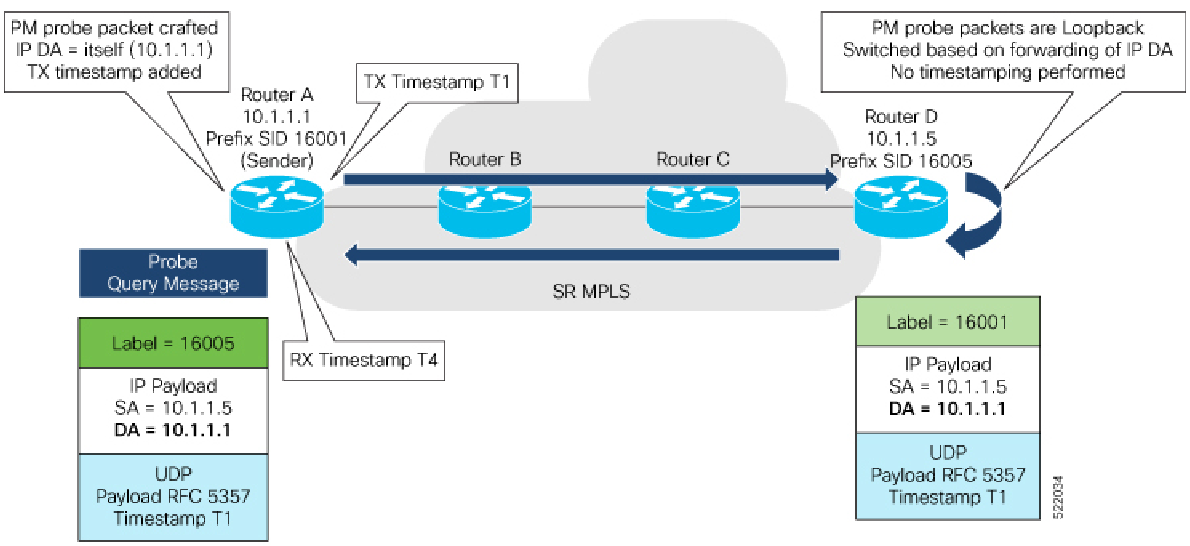
   The received timestamp (T4) stored.

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

Running Configuration

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

Verification

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

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

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

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

SR Policy Liveness Monitoring

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

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
SR Policy Liveness Monitoring	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is now supported on Cisco 8711-48Z-M routers.

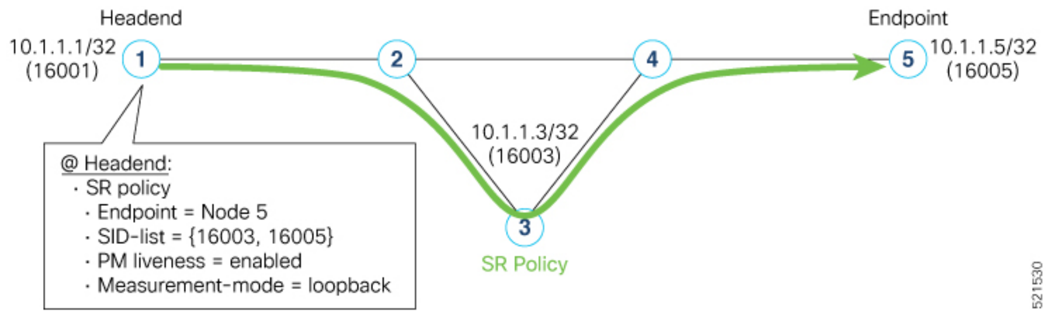
Feature Name	Release Information	Feature Description
SR Policy Liveness Monitoring	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) This feature is now supported on the Cisco 8011-4G24Y4H-I routers.
SR Policy Liveness Monitoring	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 8712-MOD-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
SR Policy Liveness Monitoring	Release 7.5.2	This feature allows you to verify end-to-end traffic forwarding over an SR Policy candidate path by periodically sending performance monitoring packets.

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

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

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

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



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

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

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

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

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

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

- **C**: The end-point node receives the PM probe packets.

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

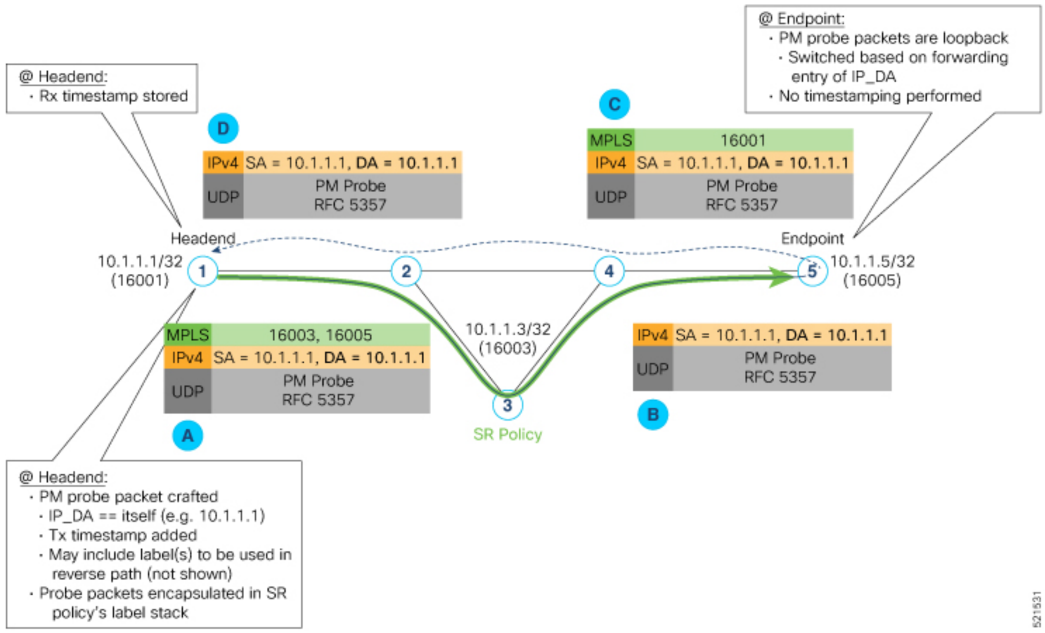
  If the head-end node inserted label(s) for the reverse path, then the packets are switched back at the end-point node based on the forwarding entry associated with the top-most reverse path label.

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

  A received (Rx) timestamp stored.

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

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



Usage Guidelines and Limitations

The following usage guidelines and limitations apply:

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

Configuring SR Policy Liveness Monitoring

Configuring SR Policy liveness monitoring involves the following steps:

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

Configuring Performance Measurement Liveness Profile

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

- **liveness-profile** {**sr-policy default** | **name** *name*}
- **probe**: Configure the probe parameters.
- **measurement-mode**: Liveness detection must use loopback mode (see Measurement Mode topic within this guide).
- **tx-interval**: Interval for sending probe packet. The default value is 3000000 microseconds and the range is from 3300 to 15000000 microseconds.
- **tos dscp** *value*: The default value is 48 and the range is from 0 to 63. You can modify the DSCP value of the probe packets, and use this value to 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 (tx-interval * multiplier).

Enabling Liveness Monitoring under SR Policy

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

- **liveness-detection**: Enables end-to-end SR Policy Liveness Detection for all segment-lists of the active and standby candidate-path that are in the forwarding table.
- **liveness-profile name** *name*: Specifies the profile name for named profiles.
- **invalidation-action {down | none}**:
  - **Down (default)**: When the PM liveness session goes down, the candidate path is immediately operationally brought down.
  - **None**: When the PM liveness session goes down, no action is taken. If logging is enabled, the failure is logged but the SR Policy operational state is not modified.
- **logging session-state-change**: Enables Syslog messages when the session state changes.
- **reverse-path label** {*BSID-value* | *NODE-SID-value* | *ADJACENCY-SID-value*}: Specifies the MPLS label to be used for the reverse path for the reply. If you configured liveness detection with ECMP hashing, you must specify the reverse path. The default reverse path uses IP Reply.
  - *BSID-value*: The Binding SID (BSID) label for the reverse SR Policy. (This is practical for manual SR policies with a manual BSID.)

- *NODE-SID-value*: The Node SID is a segment type that represents the ECMP-aware shortest path to reach a particular IP prefix from any IGP topology location
- *ADJACENCY-SID-value*: The absolute SID label of the (local) Sender Node to be used for the reverse path for the reply.

Configuration Examples

Configure a Default SR-Policy PM Liveness-Profile

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

```
RP/0/RSP0/CPU0:ios(config)# performance-measurement
RP/0/RSP0/CPU0:ios(config-perf-meas)# liveness-profile sr-policy default
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy)# probe

RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# tx-interval 150000
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# tos dscp 52
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-probe)# exit
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy)# liveness-detection
RP/0/RSP0/CPU0:ios(config-pm-ld-srpolicy-ld)# multiplier 5
```

Running Configuration:

```
performance-measurement
 liveness-profile sr-policy default
  liveness-detection
   multiplier 5
  !
  probe
   tos dscp 52
   tx-interval 150000
  !
 !
!
end
```

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

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

```
Router(config)# performance-measurement
Router(config-perf-meas)# liveness-profile name sample-profile
Router(config-pm-ld-profile)# probe
Router(config-pm-ld-probe)# tx-interval 150000
Router(config-pm-ld-probe)# tos dscp 52
Router(config-pm-ld-probe)# exit
Router(config-pm-ld-profile)# liveness-detection
Router(config-pm-ld-profile-ld)# multiplier 5
Router(config-pm-ld-profile-ld)#commit
```

Running Configuration:

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

```
  !
 !
!
end
```

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

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

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

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

Running Configuration

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

Enable Liveness Monitoring Under SR Policy

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

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

Running Config

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

```
!
end
```

Enable Liveness Monitoring under SR Policy with Optional Parameters

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

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

Running Config

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

SR Policy Liveness Monitoring - Hardware Offloading

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.

Table 5: Feature History Table

Feature Name	Release	Description
SR Policy Liveness Monitoring - Hardware Offloading	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100]) (select variants only*) *This feature is supported on: • 8711-48Z-M • 8011-12G12X4Y-A • 8011-12G12X4Y-D

Feature Name	Release	Description
SR Policy Liveness Monitoring - Hardware Offloading	Release 25.1.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100])(select variants only*); Modular Systems (8800 [LC ASIC: P100]) The process of liveness monitoring in performance measurement can now be offloaded to the router's dedicated hardware, which is the Network Processing Unit (NPU). This feature helps optimize and scale the measurement operation, enabling the operators to meet delay-bound Service Level Agreements (SLAs). * This feature is supported on: • 8011-4G24Y4H-I • 8212-48FH-M • 8711-32FH-M • 8712-MOD-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM
SR Policy Liveness Monitoring - Hardware Offloading	Release 24.4.1	This feature is now supported on: Fixed Systems (8700) (select variants only*); Centralized Systems (8600); Modular Systems (8800 [LC ASIC: Q200]); *This feature is now supported on: • 8101-32H • 8102-64H • 8201-32FH • 8202-32FH-M • 8203-88H16FH-M • 8608-RP • 88-LC0-34H14FH • 88-LC0-36FH • 88-LC0-36FH-M

Feature Name	Release	Description
SR Policy Liveness Monitoring - Hardware Offloading	Release 7.11.1	Introduced in this release on: Cisco 8011-2X2XP4L PLE Service Endpoint Router. The process of liveness monitoring in performance measurement can now be offloaded to the router's dedicated hardware, which is the Network Processing Unit (NPU). This feature helps you optimize and scale the measurement operation, enabling you meet delay-bound Service Level Agreements (SLAs). The feature introduces a new keyword **npu-offload** under the **performance-measurement liveness-profile name***liveness profile* command.

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.

Configuration Example

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

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

Running Configuration

The running configuration for this feature is as shown:

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

```
 !
!
```

Verification

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

```
Router# show performance-measurement sessions detail

Transport type : SR Policy
Measurement type : Liveness Detection
Policy name : srte_c_90005_ep_10.2.2.2
Color : 90005
Endpoint : 10.2.2.2
Instance : 7
preference : 20
Protocol-origin : Configured
Discriminator : 20
Segment-list : route_12_2
Atomic path:
Hops : 10.2.2.2
Session ID : 45
Trace ID : 3111803555
NPU Offloaded session : True
NPU number : 0
NPU session state : Session created
Retry count : 0
Last NPU notification:
Session state : Up
Timestamp : Feb 28 2023 16:28:09.411
Timestamping Enabled : True
Liveness Detection: Enabled
Session State: Up
Last State Change Timestamp: Feb 28 2023 16:28:09.411
Missed count: 0
```

Delay Measurement

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

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
Delay Measurement	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) Delay measurement in SR networks involves monitoring the end-to-end delay experienced by traffic sent over an SR policy. Link delay metrics such as average, minimum, and maximum delay, and delay variance are used to determine network latency. You can ensure compliance with Service Level Agreements (SLAs) by monitoring the end-to-end delay experienced by traffic. This feature is now supported on: • 8011-4G24Y4H-I
SR Performance Measurement: Loopback Measurement Mode	Release 7.5.2	Loopback measurement mode provides two-way and one-way measurements. PTP-capable hardware and hardware timestamping are required on the Sender but are not required on the Reflector. Liveness monitoring uses "self-addressed" PM IP packets crafted by the sender (where the destination address is the sender's own IP address); this mode of operation is referred as "loopback mode”.

The PM for delay measuremnt uses the IP/UDP packet format defined in Simple TWAMP using RFC8972 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.

Delay Measurement modes

You can measure delay using the following methods:

- One-way: Use to monitor delay experienced by data packets in a single link or path between two nodes in a network.
- Two-Way: 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.
- Loop back: Use to to monitor the end-to-end delay experienced by the traffic sent over an SR policy.

Measurement Modes

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

Feature Name	Release Information	Description
SR Performance Measurement: Loopback Measurement Mode	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) * This feature is now supported on Cisco 8711-48Z-M routers.
SR Performance Measurement: Loopback Measurement Mode	Release 7.5.2	Loopback measurement mode provides two-way and one-way measurements. PTP-capable hardware and hardware timestamping are required on the Sender but are not required on the Reflector. Liveness monitoring uses "self-addressed" PM IP packets crafted by the sender (where the destination address is the sender's own IP address); this mode of operation is referred as "loopback mode”.

Table 7: Measurement Mode Requirements

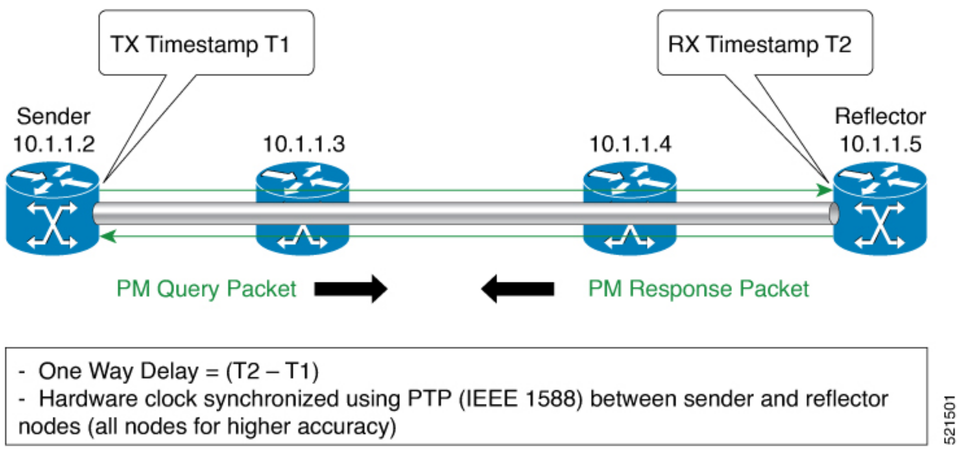
Measurement Mode	**Sender:** **PTP-Capable HW and HW Timestamping**	**Reflector:** **PTP-Capable HW and HW Timestamping**	**PTP Clock Synchronization between Sender and Reflector**
One-way	Required	Required	Required
Two-way	Required	Required	Not Required
Loopback	Required	Not Required	Not Required

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 2: One-Way



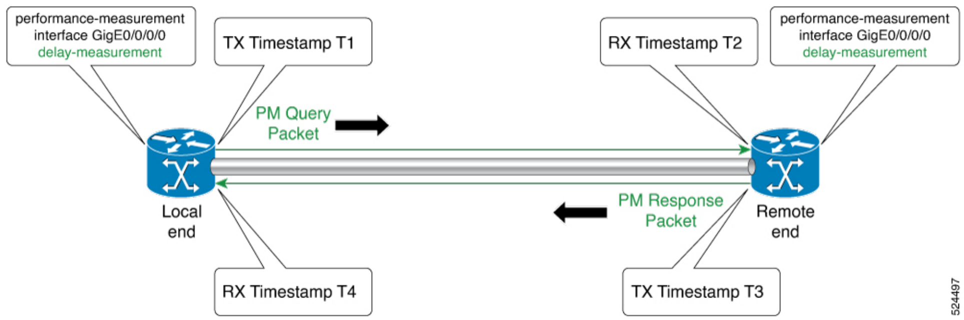
The PM query and response for one-way delay measurement can be described in the following steps:

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	Feature Description
Far-end delay metrics in one-way measurement mode	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]8700 [ASIC: K100]) (select variants only*) * This feature is supported on: • 8712-MOD-M • 8011-4G24Y4H-I
Far-end delay metrics in one-way measurement mode	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Centralized Systems (8600);Modular Systems (8800 [LC ASIC: Q100, Q200, P100])(select variants only*) 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. *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM



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.

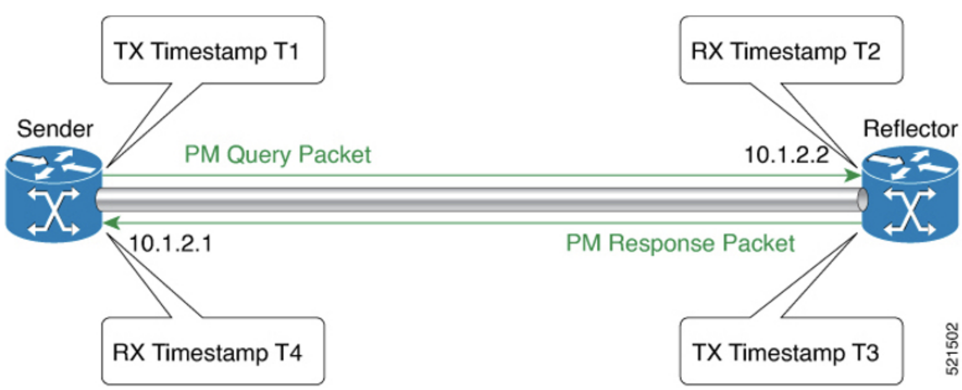
Note If the measurement mode is not set to one-way or is set to loopback, SR PM skips the far-end delay metric computation.

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 3: Two-Way



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

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

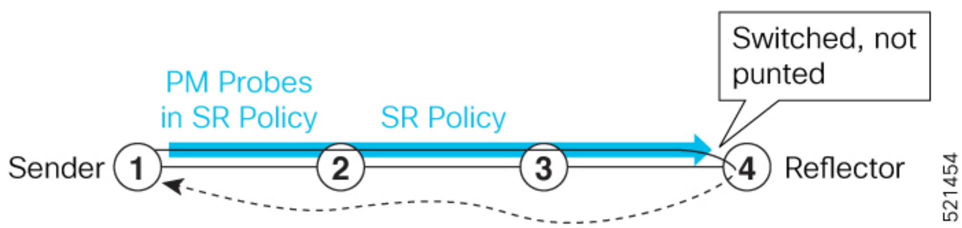
Loopback Measurement Mode

Loopback 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 4: Loopback



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

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

Link Delay Measurement

Table 9: Feature History Table

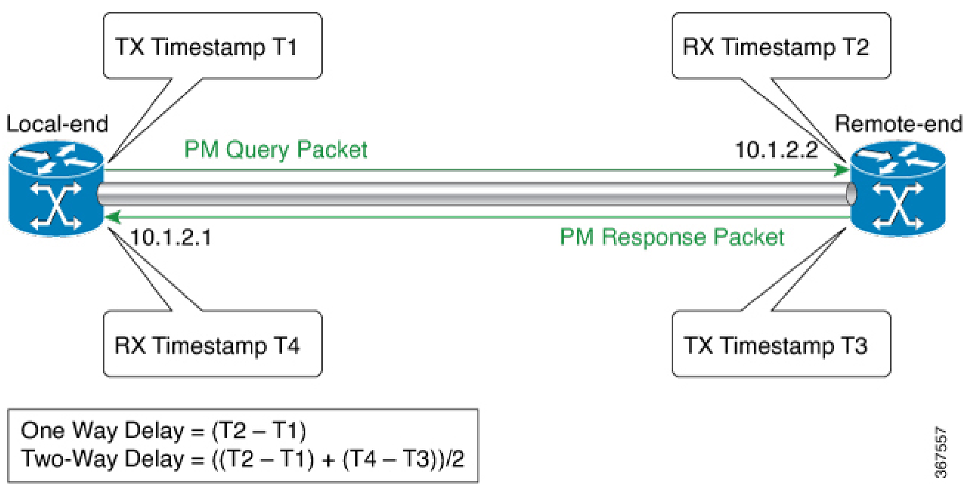
Feature Name	Release Information	Feature Description
Link Delay Measurement using TWAMP Light Encoding	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100,8010 [ASIC: A100])(select variants only*) *This feature is now supported on: • 8711-48Z-M • 8011-12G12X4Y-A • 8011-12G12X4Y-D
Link Delay Measurement using TWAMP Light Encoding	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100])(select variants only*) *This feature is supported on Cisco 8011-4G24Y4H-I routers.
Link Delay Measurement using TWAMP Light Encoding	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 8712-MOD-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

Feature Name	Release Information	Feature Description
Link Delay Measurement using TWAMP Light Encoding	Release 7.3.1	The PM for link delay uses the IP/UDP packet format defined in RFC 8972 (Simple TWAMP) for probes. Two-Way Active Measurement Protocol (TWAMP) adds two-way or round-trip measurement capabilities. TWAMP employs time stamps applied at the echo destination (reflector) to enable greater accuracy.

The PM for link delay uses the IP/UDP packet format defined in RFC 8972 (simple TWAMP) for probes. Two-Way Active Measurement Protocol (TWAMP) adds two-way or round-trip measurement capabilities. TWAMP employs time stamps applied at the echo destination (reflector) to enable greater accuracy. In the case of TWAMP Light, the Session-Reflector doesn't necessarily know about the session state. The Session-Reflector simply copies the Sequence Number of the received packet to the Sequence Number field of the reflected packet. The controller receives the reflected test packets and collects two-way metrics. This architecture allows for collection of two-way metrics.

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

Figure 5: Performance Measurement for Link Delay



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

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

Restrictions and Usage Guidelines for PM for Link Delay

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

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

Configuration Example: PM for Link Delay

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

- **probe measurement mode**: The default measurement mode for probe is two-way delay measurement. If you are configuring one-way delay measurement, hardware clocks must be synchronized between the local-end and remote-end routers using precision time protocol (PTP).
- **protocol**: Interface delay measurement uses RFC 8972 (simple TWAMP) with IP/UDP encapsulation.
- **tx-interval**: Interval for sending probe packet. The default value is 3000000 microseconds and the range is from 3300 to 15000000 microseconds.
- **computation interval**: Interval for metric computation. Default is 30 seconds; range is 1 to 3600 seconds.
- **periodic advertisement**: Periodic advertisement is enabled by default.
- **periodic-advertisement interval**: The default value is 120 seconds and the interval range is from 30 to 3600 seconds.
- **periodic-advertisement threshold**: The default value of periodic advertisement threshold is 10 percent.
- **periodic-advertisement minimum change**: The default value is 1000 microseconds (usec) and the range is from 0 to 10000 microseconds.
- **accelerated advertisement**: Accelerated advertisement is disabled by default.
- **accelerated-advertisement threshold**: The default value is 20 percent and the range is from 0 to 100 percent.
- **accelerated-advertisement minimum change**: The default value is 1000 microseconds and the range is from 1 to 100000 microseconds.

```
RP/0/0/CPU0:router(config)# performance-measurement delay-profile interfacesdefault
RP/0/0/CPU0:router(config-pm-dm-intf)# probe
RP/0/0/CPU0:router(config-pm-dm-intf-probe)# measurement-mode one-way
RP/0/0/CPU0:router(config-pm-dm-intf-probe)# tx-interval 30000
```

```
RP/0/0/CPU0:router(config-pm-dm-intf-probe)# exit

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

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

Configure the UDP Destination Port

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

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

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

This example shows how to configure the UDP destination port.

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

Enable PM for Link Delay Over an Interface

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

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

Verification

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

--------------------------------------------------------------------------------
0/0/CPU0
--------------------------------------------------------------------------------
Interface Delay-Measurement:
  Profile configuration:
    Measurement Type                                   : Two-Way
    Probe computation interval                         : 30 (effective: 30) seconds
    Type of services                                   : Traffic Class: 6, DSCP: 48
    Burst interval                                     : 3000 (effective: 3000) mSec
    Burst count                                        : 10 packets
```

```
    Encap mode                                    : UDP
    Payload Type                                  : TWAMP-light
    Destination sweeping mode                     : Disabled
    Periodic advertisement                        : Enabled
      Interval                                    : 120 (effective: 120) sec
      Threshold                                   : 10%
      Minimum-Change                              : 500 uSec
    Advertisement accelerated                     : Disabled
    Threshold crossing check                      : Minimum-delay

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

Thu Dec 12 14:09:59.162 PST

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


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

SR Policy Delay-Measurement:
  Total active sessions                           : 0
  Counters:
    Packets:
      Total sent                                  : 0
      Total received                              : 0
    Errors:
        TX:
          Reason interface down                   : 0
```

```
          Reason no MPLS caps                : 0
          Reason no IP address               : 0
          Reason other                       : 0
        RX:
          Reason negative delay              : 0
          Reason delay threshold exceeded    : 0
          Reason missing TX timestamp        : 0
          Reason missing RX timestamp        : 0
          Reason probe full                  : 0
          Reason probe not started           : 0
          Reason control code error          : 0
          Reason control code notif          : 0
    Probes:
      Total started                          : 0
      Total completed                        : 0
      Total incomplete                       : 0
      Total advertisements                   : 0

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

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

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

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

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

Interface Name: GigabitEthernet0/2/0/0 (ifh: 0x1004060)
  Delay-Measurement                    : Enabled
  Loss-Measurement                     : Disabled
```

```
Configured IPv4 Address          : 10.10.10.2
Configured IPv6 Address          : 10:10:10::2
Link Local IPv6 Address          : fe80::3a:6fff:fec9:cd6b
Configured Next-hop Address      : Unknown
Local MAC Address                : 023a.6fc9.cd6b
Next-hop MAC Address             : 0291.e460.6707
Primary VLAN Tag                 : None
Secondary VLAN Tag               : None
State                            : Up

Delay Measurement session:
  Session ID          : 1

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

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

  Current Probe:
    Started at Dec 12 2019 14:15:43.154 (26.766 seconds ago)
    Packets sent: 9, received: 9, lost: 0
    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.
show performance-measurement responder counters [**interface** *interface*] [**location** *location-name*]	Displays the PM link-delay session counters on the remote-end router.

SR Performance Measurement Named Profiles

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

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

Delay Profile

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

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

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

Apply the delay profile for an SR Policy.

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

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

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

Running Configuration

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

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

Verification

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

----------------------------
0/RSP0/CPU0
----------------------------
SR Policy Delay Measurement Profile Name: profile2
  Profile configuration:
    Measurement mode                            : One-way
    Protocol type                               : TWAMP-light
    Encap mode                                  : UDP
    Type of service:
      PM-MPLS traffic class                     : 6
      TWAMP-light DSCP                          : 63
    Probe computation interval                  : 60 (effective: 60) seconds
    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 name profile3
Router(config-pm-ld-profile)# probe
Router(config-pm-ld-probe)# tx-interval 60000
Router(config-pm-ld-profile)# probe
Router(config-pm-ld-probe)# tx-interval 60000
Router(config-pm-ld-probe)# tos dscp 10
Router(config-pm-ld-probe)# exit

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

Apply the liveness profile for the SR policy

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

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

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

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

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

Running Configuration

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

segment-routing
 traffic-eng
  policy TRST2
   color 40 end-point ipv4 20.20.20.20
   candidate-paths
    preference 50
     explicit segment-list LIST3
      weight 2
```

```
      !
      explicit segment-list LIST4
       weight 3
      !
     !
    !
    performance-measurement
     liveness-detection
      liveness-profile name profile3
     !
```

Verification

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

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

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

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

On-Demand SR Policy

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

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

Running Configuration

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

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

Verification

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

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

Delay Normalization

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

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

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

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

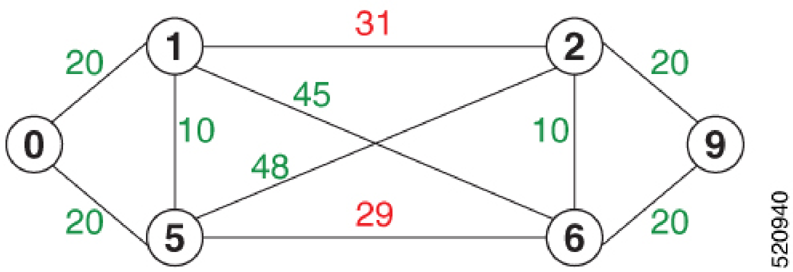
The following formula is used to calculate the normalized value:

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

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

Example

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



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

- Interval = 10
- Offset = 3

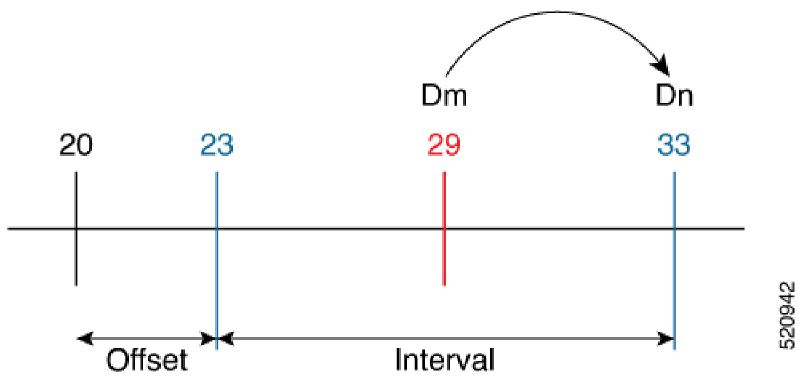
The measured delays will be normalized as follows:

- **Dm** = 29

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

  b = 2 * 10 + 3 = 23

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

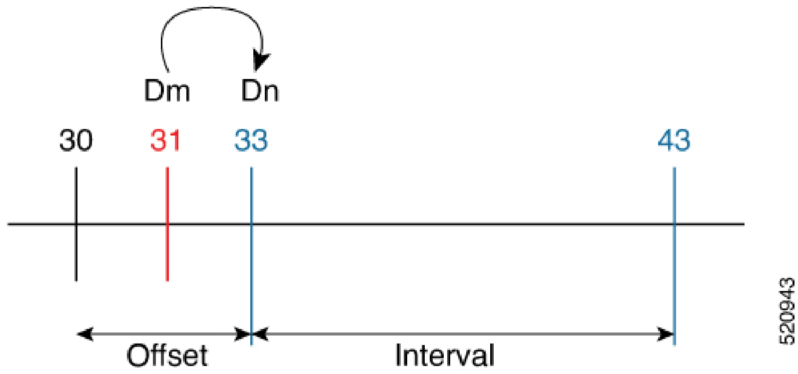


- **Dm** = 31

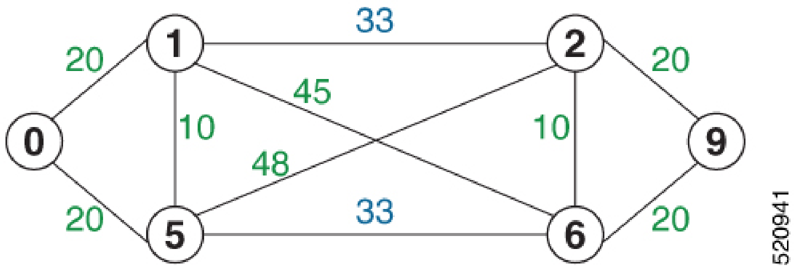
  a = 31 / 10 = 3 (3.1, rounded down to 3)

  b = 3 * 10 + 3 = 33

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



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



Configuration

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

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

IS-IS Configuration

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

OSPF Configuration

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

Link Anomaly Detection with IGP Penalty

Table 10: Feature History Table

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

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

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

Usage Guidelines and Limitations

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

Configuration Example

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

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

Running Configuration

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

Delay Measurement for IP Endpoint

Table 11: Feature History Table

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

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

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

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

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

Collecting IP Endpoint Probe Statistics

- Statistics associated with the probe for delay metrics are available via Histogram and Streaming Telemetry.
- Model Driven Telemetry (MDT) is supported for the following data:
  - Summary, endpoint, session, and counter show command bags.
  - History buffers data
- Model Driven Telemetry (MDT) and Event Driven Telemetry (EDT) are supported for the following data:
  - Delay metrics computed in the last probe computation-interval (event: probe-completed)
  - Delay metrics computed in the last aggregation-interval; that is, end of the periodic advertisement-interval (event: advertisement-interval expired)
  - Delay metrics last notified (event: notification-triggered)
- The following xpaths for MDT/EDT is supported:
  - `Cisco-IOS-XR-perf-meas-oper:performance-measurement/nodes/node/endpoints/endpoint-delay/endpoint-last-probes`
  - `Cisco-IOS-XR-perf-meas-oper:performance-measurement/nodes/node/endpoints/endpoint-delay/endpoint-last-aggregations`
  - `Cisco-IOS-XR-perf-meas-oper:performance-measurement/nodes/node/endpoints/endpoint-delay/endpoint-last-advertisements`

Supported Features

- IPv6 Endpoint Delay in Default VRF (over SRv6)
- SRv6 Endpoint Delay in Default VRF (Endpoint can be Node SID, Flex-Algo SID, Packed uSID carrier)
- IPv6 Endpoint Delay in VRF (static uDT6)
- IPv6 Endpoint Delay in VRF (dynamic uDT6 encap)
- IPv4 Endpoint Delay in VRF or GRT (static uDT4)
- IPv4 Endpoint Delay in VRF or GRT (dynamic uDT4 encap)

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.

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

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

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

SR Policy End-to-End Delay Measurement

Feature Name	Release	Feature Description
SR Policy End-to-End Delay Measurement	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8711-48Z-M routers.

Feature Name	Release	Feature Description
SR Policy End-to-End Delay Measurement	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) This feature is now supported on: • 8011-4G24Y4H-I
SR Policy End-to-End Delay Measurement	Release 24.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100]) (select variants only*) * This feature is supported on Cisco 8712-MOD-M routers.
SR Policy End-to-End Delay Measurement	Release 7.5.2	This feature allows you to monitor the end-to-end delay experienced by the traffic sent over an SR policy to ensure that the delay does not exceed the requested "upper-bound" and violate SLAs.

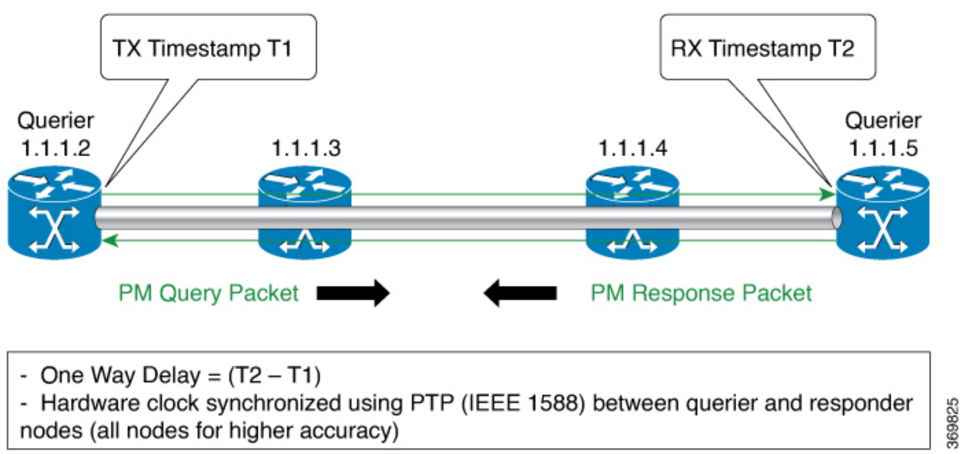
The PM for SR Policy uses IP/UDP packet format defined in RFC 8972 (Simple TWAMP) for probes.

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

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

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

Figure 6: Performance Measurement for SR Policy End-to-End Delay



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

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

Restrictions and Usage Guidelines for PM for SR Policy Delay

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

Enable One-Way Delay Mode

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

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

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

Configuring Performance Measurement Parameters

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

- **probe**: The default mode for probe is one-way delay measurement.
- **tx-interval**: Interval for sending probe packet. The default value is 3000000 microseconds and the range is from 3300 to 15000000 microseconds.
- **computation interval**: Interval for metric computation. Default is 30 seconds; range is 1 to 3600 seconds.
- **protocol**:
  - **twamp-light**: SR Policy delay measurement using simple TWAMP RFC 8972 with IP/UDP encap. This is the default protocol.
- **tos**: Type of Service
  - **dscp** *value*: The default value is 0 and the range is from 0 to 63.
  - **traffic-class** *value*: The default value is 0 and the range is from 0 to 7.
- **advertisement threshold-check**: The advertisement threshold-check has three types:
  - **average-delay**: Enable average-delay threshold-check.
  - **maximum-delay**: Enable maximum-delay threshold-check.
  - **minimum-delay**: Enable minimum-delay threshold-check.

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

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

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

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

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

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

Configure the UDP Destination Port

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

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

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

This example shows how to configure the UDP destination port.

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

Enable Performance Measurement for SR Policy

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

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

SR Policy Probe IP/UDP ECMP Hashing Configuration

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

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

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

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

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

Fallback delay advertisement

Table 12: Feature History Table

Feature Name	Release Information	Feature Description
Fallback delay advertisement	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: K100])(select variants only*) *This feature is supported on: • 8712-MOD-M • 8011-4G24Y4H-I

Feature Name	Release Information	Feature Description
Fallback delay advertisement	Release 24.4.1	

Feature Name	Release Information	Feature Description
		Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Centralized Systems (8600); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]))(select variants only*) 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. *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM The feature introduces these changes: **CLI:** The **performance-measurement interface** command is modified with a new **advertise-delay fallback** keyword. **YANG Data Models:** • Cisco-IOS-XR-um-performance-measurement-cfg.yang See (GitHub, Yang Data Models

Feature Name	Release Information	Feature Description
		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 13: Feature History Table

Feature Name	Release Information	Feature Description
Two-Way Active Measurement Protocol Light Source Address Filtering	Release 25.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8711-48Z-M routers.
Two-Way Active Measurement Protocol Light Source Address Filtering	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) This feature is now supported on Cisco 8011-4G24Y4H-I routers.
Two-Way Active Measurement Protocol Light Source Address Filtering	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 8712-MOD-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

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-um-performance-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

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.

Table 14: Feature History Table

Feature Name	Release	Description
Synthetic Loss Measurement	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) This feature is now supported on the Cisco 8011-4G24Y4H-I routers.
Synthetic Loss Measurement	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 8712-MOD-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM

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 TWAMP test packets that are deployed in the delay profile are used to simulate network traffic, test network performance, and evaluate network infrastructure. They are 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 8000 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 sent: 40, received: 40, lost: 0
      ...
    Next advertisement:
      ...
      Packets sent: 10, received: 10, lost: 0
      …
    Current computation:
      ...
      Packets sent: 4, received: 4, lost: 0
```

```
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 sent: 40, received: 40, lost: 0
...
    Next advertisement:
     ...
      Packets sent: 30, received: 30, lost: 0
      ...
    Current computation:
     ...
      Packets sent: 6, received: 6, lost: 0
      …
```

```
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 sent: 40, received:40, lost: 0
      ...
    Next advertisement:
      ...
      Packets sent: 10, received:10, lost: 0
      ...
    Last computation:
      Packets sent: 10, received:10, lost: 0
      ...
    Current computation:
      ...
      Packets sent: 6, received: 6, lost: 0
      …
…

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 sent: 88, received: 88, lost: 0
...
    Next advertisement:
      ...
      Packets sent: 132, received: 132, lost: 0
     ...
    Last computation:

      Packets sent: 4, received: 4, lost: 0
      ...
    Current computation:
      ...
      Packets sent: 4, received: 4, lost: 0
      ...
    Segment-List              : R4
      ...
      Last advertisement:
        ...
        Packets sent: 88, received: 88, lost: 0
        ...
      Next advertisement:
        ...
        Packets sent: 132, received: 132, lost: 0
        ...
      Last computation:
        Packets sent: 4, received: 4, lost: 0
        ...
      Current computation:
        Packets sent: 4, received: 4, lost: 0
```

…

```
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 sent: 30, received: 30, lost: 0
    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 sent: 30, received: 30, lost: 0
    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 15: Feature History Table

Feature Name	Release Information	Feature Description
Delay and synthetic loss measurement for GRE tunnel interfaces	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100])(select variants only*) *This feature is supported on Cisco 8712-MOD-M routers.

Feature Name	Release Information	Feature Description
Delay and synthetic loss measurement for GRE tunnel interfaces	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: Q100, Q200, P100])(select variants only*) 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. *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 88-LC1-36EH • 88-LC1-12TH24FH-E • 88-LC1-52Y8H-EM 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 18 and Synthetic Loss Measurement, on page 55 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

Limitations

The following limitations apply:

- Loopback delay measurement mode does not support GRE interfaces.
- Liveness measurement does not support GRE interfaces.
- GRE over SRv6 or SR-MPLS underlay is not supported for performance measurement.
- IP-in-IP tunnel mode (without the GRE header) is not supported.
- Interface delay measurement is not supported in nondefault VRF.
- The Cisco 8000 series routers do not support IPv6 over GRE tunnels.
- The Cisco 8000 series routers do not support performance measurement involving GRE tunnel interfaces on ETM-enabled line cards.
- Cisco 8000 Series Modular Routers with line cards with Q200 ASIC does not support SR-PM over GRE tunnels.
- The routers do not support IPv6 over GRE tunnels on the responder node.

Usage guidelines

The following guidelines apply:

- You can measure only one path for GRE tunnels with ECMP paths.
- GRE encapsulation mode for IP tunnel is supported.

- GRE interface tunnels over an IGP network are supported.

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 57 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
```

Link residual bit error rate measurement using STAMP

Link residual bit error rate measurement is a performance monitoring mechanism that

- uses the Simple Two-way Active Measurement Protocol (STAMP) with extra padding type-length-values (TLVs) and configurable bit patterns to provide active, real-time error detection
- measures both bit error rate (BER) and packet error rate (PER) in the forward and reverse directions, and
- reports metrics in parts per million (PPM) to identify link degradations that standard error correction does not detect

Table 16: Feature History Table

Feature Name	Release Information	Feature Description
Link residual bit error rate measurement using STAMP	Release 26.1.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q200, P100], 8700 [ASIC: P100, K100], 8010 [ASIC: A100]); Centralized Systems (8600 [ASIC: Q200]); Modular Systems (8800 [LC ASIC: Q200, P100]) You can help prevent packet loss and service disruptions by detecting the physical layer issues early and troubleshooting them quickly. This capability is achieved by actively measuring residual bit errors and reporting metrics to identify link degradations. This feature introduces or modifies the following changes: CLI: • **extra-padding** • **anomaly-ber** keyword is added in **performance-measurement delay-profile** command. • The output of the **show performance-measurement** is modified to include BER detection details.

Key concepts of link residual bit error rate measurement

- Bit error rate (BER): The ratio of incorrectly received bits (error bits) to the total number of bits transmitted during a specific time interval. BER is also referred to as residual BER.

  BER (in PPM) = (Total bit error count in the padding TLV / Total bits in the padding TLV) × 1,000,000

- Packet rate with bit error (PER): Tracks and reports the number of probe packets with bit errors, the number of probe packets transmitted, and the error rate for each interval. PER is calculated as:

```
Packet error rate (PER) = (total number of packets with at least one bit error) / (total
number of packets received within an interval)
```

- Extra padding TLV (Type 1): A STAMP extension defined in RFC 8972 that is used to increase the packet size to the link MTU and carry the bit pattern used for error detection.
- Simple Two-way Active Measurement Protocol (STAMP): A protocol that enables active performance measurements by exchanging timestamped probe packets between a sender and a reflector.
- Padding size: The number of bytes added to a STAMP test packet using the extra padding data TLV.
- Padding pattern: The bit sequence used to populate the extra padding data TLV for error detection.
- Padding validation: A mechanism that enables bit-level comparison of received padding data against the expected pattern.

BER calculation

- Configured padding size: 1000 bytes
- Total bits transmitted in padding: 1000 bytes × 8 bits = 8000 bits
- Bit errors detected by the reflector: 4 bits

BER (in PPM) = (4 / 8000) × 1,000,000 = 500 PPM

How performance measurement for link bundle members work

Summary

The process involves initializing unique micro-sessions for each bundle member and tagging probes with specific member IDs to enforce physical pathing. The reflector ensures symmetric return of the probes, allowing the sender to aggregate and report independent performance data for every physical link in the bundle.

Workflow

These stages describe how performance measurement for link bundle members works.

1. Initializing micro-sessions: The user manually configures a STAMP session for each member interface within the bundle. The sender then allocates a unique STAMP session ID for each configured member link.
2. Tagging the probe: The sender inserts a micro-session ID TLV (Type 11) into the STAMP packet. This TLV carries the 16-bit member ID required to identify the specific physical link.
3. Enforcing the path: The sender pre-routes the probe packet to ensure it is transmitted over the specific physical member link associated with the session.
4. Symmetric reflecting: The reflector receives the probe and identifies the incoming physical interface. It uses the Micro-session ID TLV to ensure the reflected reply is transmitted back over the exact same physical member link.
5. Aggregating per-member data: The sender collects the reflected probes for each configured micro-session and computes independent delay, loss, and BER metrics for each member link with an active session.

Guidelines for configuring BER measurement

Follow these guidelines to configure BER measurement:

- You need to configure BER detection under the `extra-padding` sub-mode.
- BER detection is disabled unless you explicitly enable it.

  You can configure a pattern without configuring a size, but the router does not apply the pattern or display it in operational output unless BER detection is enabled. The default pattern is `0xFF00`.
- When you enable the `local` option, the sender omits the BER detection TLV from the packet. The reflector uses the default pattern if no explicit pattern is configured.
- Errors detected by hardware mechanisms, such as Ethernet CRC errors and checksum errors, are not reported by the BER feature. In these cases, packets are dropped. The BER feature identifies errors that hardware mechanisms do not capture.

Configure link residual BER measurement

Use this task to enable link residual BER measurement using STAMP.

You enable link residual bit error rate (BER) measurement by configuring STAMP extra padding with validation in a performance-measurement delay profile and applying that profile to an interface. If you use a local padding pattern, you must also configure the expected pattern on the reflector.

Procedure

Step 1 Enable BER detection on the sender.

Configure the extra padding TLV size, pattern, and validation. You can configure BER anomaly, advertisement and logging as well on the sender. You need to configure BER detection under the `extra-padding` sub-mode.

Note
If you use a local pattern, configure the same pattern on both the sender and the reflector.

Example:

```
Router(config)# performance-measurement
Router(config-perf-meas)#delay-profile interfaces default
Router(config-pm-dm-intf)#probe
Router(config-pm-dm-intf-probe)#extra-padding
Router(config-pm-intf-profile-padding)#size 1500
Router(config-pm-intf-profile-padding)#pattern 0xDE local
Router(config-pm-intf-profile-padding)#ber-detection
Router(config-pm-intf-profile-padding)#exit
```

Configure advertisement and logging on the sender.

```
Router(config-pm-dm-intf)#advertisement
Router(config-pm-dm-intf-adv)#periodic interval 120
Router(config-pm-dm-intf-adv)#accelerated
Router(config-pm-dm-intf-adv-acc)#ber-threshold 10000
Router(config-pm-dm-intf-adv-acc)#exit
```

```
Router(config-pm-dm-intf-adv)#logging
Router(config-pm-dm-intf-adv-log)#
```

(optional) Configure the anomaly detection on the sender.

```
Router(config)# performance-measurement
 Router(config-perf-meas)#delay-profile interfaces default
 Router(config-pm-dm-intf)#advertisement
 Router(config-pm-dm-intf-adv)#anomaly-loss upper-bound 1 lower-bound 45
 Router(config-pm-dm-intf-adv)#anomaly-ber upper-bound 3 lower-bound 7
 Router(config-pm-dm-intf-adv)#anomaly-per upper-bound 2 lower-bound 4
```

Step 2 Apply the profile to the interface on the sender.

Example:

```
Router#configure
Router(config)#performance-measurement
Router(config-perf-meas)#interface HundredGigE 0/0/0/16
Router(config-pm-intf)#delay-measurement
Router(config-pm-intf-dm)#delay-profile name foo
```

Step 3 Configure the expected pattern on the reflector.

If you enable the local pattern option on the sender, you must configure the expected pattern on the reflector.

When you enable the `local` option, the sender omits the BER detection TLV from the packet. The reflector uses the default pattern if no explicit pattern is configured.

Example:

```
Router(config-perf-meas)#protocols stamp
Router(config-pm-protocol-stamp)#stamp-reflector
Router(config-pm-proto-stamp-reflector)#extra-padding
Router(config-pm-proto-stamp-reflector-padding)#expected-pattern 0xDE
Router(config-pm-proto-stamp-reflector-padding)#
```

Step 4 Verify the link residual BER measurement.

Example:

```
Router#show performance-measurement int HundredGigE 0/0/0/16 detail

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

Interface Name: HundredGigE 0/0/0/16 (ifh: 0x7800355c)
  Description                      : Not set
  Parent interface                 : None
  Delay-Measurement                : Enabled
  Loss-Measurement                 : Enabled
  Path-Tracing                     : Disabled
  Configured IPv4 Address          : 10.12.1.1
  Configured IPv6 Address          : ::
  Link Local IPv6 Address          : fe80::1
  Configured Next-hop Address      : Unknown
  Local MAC Address                : axxxx.buuu.a666
  Next-hop MAC Address             : 2xxx.967c.6xxx
  In-use Source Address            : 10.12.1.1
  In-use Destination Address       : 10.12.1.2
  Primary VLAN Tag                 : None
  Secondary VLAN Tag               : None
  State                            : Up
  Ethernet MTU                     : 9200
```

```
  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: Feb 05 2026 15:35:45.076 (603896.300 seconds ago)
      Advertised reason: Advertise delay config
      Advertised delays (uSec): avg: 10, min: 10, max: 10, variance: 0
      Advertised BER: bit errors: invalid, far-end bit errors: invalid, total bits: 0
      Packets sent: 0, received: 0, lost: 0, with bit errors: invalid, with far-end bit errors:
invalid
      Min-Max A flag set: False
      Anomaly-Loss A flag set: False
      BER anomaly flag set: False
      Far-end BER anomaly flag set: False
      PER anomaly flag set: False
      Far-end PER anomaly flag set: False

    Next advertisement:
      Threshold check scheduled in 2 more probes (roughly every 120 seconds)
      Aggregated delays (uSec): avg: 1, min: 1, max: 1, variance: 0
      Rolling average (uSec): 1
      Aggregated BER: bit errors: 60000, far-end bit errors: 60000, total bits: 120000
      Packets sent: 10, received: 10, lost: 0, with bit errors: 10, with far-end bit errors: 10

    Last computation:
      Packets sent: 10, received: 10, lost: 0, with bit errors: 10, with far-end bit errors: 10
      Measured delays (uSec): avg: 1, min: 1, max: 1, variance: 0
      Measured BER: bit errors: 60000, far-end bit errors: 60000, total bits: 120000
      Min-Max A flag set: False
      Anomaly-Loss A flag set: False
      BER anomaly flag set: False
      Far-end BER anomaly flag set: False
      PER anomaly flag set: False
      Far-end PER anomaly flag set: False

    Current computation:
      Started at: Feb 12 2026 15:20:37.556 (3.820 seconds ago)
      Packets sent: 2, received: 2, lost: 0, with bit errors: 2, with far-end bit errors: 2
      Measured delays (uSec): avg: 1, min: 1, max: 1, variance: 0
      Measured BER: bit errors: 12000, far-end bit errors: 12000, total bits: 24000
      Next probe scheduled at: Feb 12 2026 15:21:07.553 (in 26.177 seconds)
      Next packet will be sent in 2.177 seconds
      Packet sent every 3.0 seconds
      Responder IP            : 10.12.1.2
      Number of Hops          : 1
      Probe samples:
        Packet Rx Timestamp       Measured Delay (nsec)
        Feb 12 2026 15:20:13.560            850
        Feb 12 2026 15:20:16.562            850
        Feb 12 2026 15:20:19.562            849
        Feb 12 2026 15:20:22.560            849
        Feb 12 2026 15:20:25.560            849
        Feb 12 2026 15:20:28.559            845
        Feb 12 2026 15:20:31.562            855
        Feb 12 2026 15:20:34.562            853
        Feb 12 2026 15:20:37.561            844
        Feb 12 2026 15:20:40.560            849
      Probe far-end samples:
        No History
      Probe bit error samples:
```

```
Packet Rx Timestamp        Bit Err  Far-end Bit Err  Total Bits
Feb 12 2026 15:20:13.560      6000             6000       12000
Feb 12 2026 15:20:16.562      6000             6000       12000
Feb 12 2026 15:20:19.562      6000             6000       12000
Feb 12 2026 15:20:22.560      6000             6000       12000
Feb 12 2026 15:20:25.560      6000             6000       12000
Feb 12 2026 15:20:28.559      6000             6000       12000
Feb 12 2026 15:20:31.562      6000             6000       12000
Feb 12 2026 15:20:34.562      6000             6000       12000
Feb 12 2026 15:20:37.561      6000             6000       12000
Feb 12 2026 15:20:40.560      6000             6000       12000
Note: "-" stands for invalid measurement

Router# show performance-measurement interface brief HundredGigE 0/0/0/16
Interface Name           Tx/Rx/Err/FE-Err       Avg/Min/Max/Variance (uSec)    Bit Err/FE-Err/Total
----------------------------------------------------------------------------------------------------
HundredGigE 0/0/0/16      3/3/3/3               707/669/737/38                   24/24/96
```

Performance measurement for link bundle members

The performance measurement for link bundle members is a monitoring feature that

- tracks delay, loss, and bit error rate (BER) metrics for individual physical links within a bundle
- utilizes STAMP protocol with the micro-session ID TLV to ensure probe traffic remains on the designated physical path, and
- supports independent profile configurations for each member link to allow for varying link characteristics

Table 17: Feature History Table

Feature Name	Release Information	Feature Description
Performance measurement for link bundle members	Release 26.1.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q200, P100], 8700 [ASIC: P100], 8010 [ASIC: A100]); Centralized Systems (8600 [ASIC: Q200]); Modular Systems (8800 [LC ASIC: Q200, P100]) You can enhance fault isolation, ensure greater bandwidth availability, and boost overall network reliability and performance for each individual bundle member by measuring delay, loss, and bit errors on each link using STAMP. This capability allows you to detect and isolate link issues at the member level.

Key Concepts of link bundle members

- STAMP: STAMP is a network protocol used to measure network performance metrics such as delay, packet loss, and jitter between two endpoints. It operates by sending probes between a sender and a receiver to collect and report performance data.
- Micro-session: A specific STAMP session established for an individual physical member of a bundle. Each micro-session is identified by a unique Session ID.

- Micro-session ID TLV (Type 11): A STAMP extension (RFC 9534) that carries the 16-bit member ID, allowing the sender and reflector to identify which physical link the probe belongs to

Challenges of aggregate bundle monitoring

Link Aggregation Groups (LAG) or bundles are often treated as a single logical entity, however, the individual physical members within a bundle may follow diverse physical paths or utilize different optical components. A single underperforming member link experiencing high latency or intermittent bit errors can degrade the performance of the entire bundle, often in a way that is difficult to troubleshoot using aggregate metrics. This feature provides per-member visibility, enabling operators to identify, isolate, and remove problematic member links from a bundle to maintain consistent service levels.

Restriction for link bundle members

Interface types

- Layer 2 bundles are not supported because the protocol requires an IP address for communication.
- Performance measurement for bundles of VLAN links is not supported.
- You cannot enable this feature on sub-interfaces.

Measurement limitations

- Measurement sessions are CPU-hosted only. Hardware (NPU) offload is not supported.
- If you shut down the main bundle interface, the member performance measurement sessions will stop.

How performance measurement for link bundle members work

Summary

The process involves initializing unique micro-sessions for each bundle member and tagging probes with specific member IDs to enforce physical pathing. The reflector ensures symmetric return of the probes, allowing the sender to aggregate and report independent performance data for every physical link in the bundle.

Workflow

These stages describe how performance measurement for link bundle members works.

1. Initializing micro-sessions: The user manually configures a STAMP session for each member interface within the bundle. The sender then allocates a unique STAMP session ID for each configured member link.
2. Tagging the probe: The sender inserts a micro-session ID TLV (Type 11) into the STAMP packet. This TLV carries the 16-bit member ID required to identify the specific physical link.
3. Enforcing the path: The sender pre-routes the probe packet to ensure it is transmitted over the specific physical member link associated with the session.

4. Symmetric reflecting: The reflector receives the probe and identifies the incoming physical interface. It uses the Micro-session ID TLV to ensure the reflected reply is transmitted back over the exact same physical member link.
5. Aggregating per-member data: The sender collects the reflected probes for each configured micro-session and computes independent delay, loss, and BER metrics for each member link with an active session.

Configure performance measurement for link bundle members

Use this task to enable granular monitoring of delay, loss, and bit error rate for individual physical links within a bundle. This allows you to identify and isolate underperforming links that aggregate bundle metrics might hide.

Before you begin

Ensure that you configure the main bundle and add the members to the bundle before you configure this feature.

Procedure

Step 1 Enable delay measurement for each physical member of the bundle.

In this example, `HundredGigE 0/0/0/4/0` and `HundredGigE 0/0/0/4/1` are the member interfaces of the main bundle, `Bundle-Ether12`.

Example:

```
Router(config-pm-intf)# interface HundredGigE 0/0/0/4/0
Router (config-pm-intf)# delay-measurement
Router (config-pm-intf-dm)# exit

Router(config-pm-intf)# interface HundredGigE 0/0/0/4/1
Router (config-pm-intf)# delay-measurement
Router (config-pm-intf-dm)# exit
```

Step 2 Verify the configuration using these show commands.

Example:

```
Router#show performance-measurement interface delay brief

Interface Name          Tx/Rx/Err/FE-Err    Avg/Min/Max/Variance (uSec)    Bit Err/FE-Err/Total
--------------------------------------------------------------------------------------------------
HundredGigE0/0/0/4/0    12/12/-/-           1/1/1/0                        BER detection not enable
HundredGigE0/0/0/4/1    11/11/-/-           1/1/1/0                        BER detection not enable
```

Use the **show performance-measurement interface** command to verify that the local MAC address of the bundle members matches the parent bundle.

```
Router#show performance-measurement interfaces hundredGigE 0/0/0/4/0 detail | i Local MAC Address
  Local MAC Address                : bc9fd.xxxx.1u04

Router#show performance-measurement interfaces hundredGigE 0/0/0/4/1 detail | i Local MAC Address
  Local MAC Address                : bc9fd.xxxx.1d04
```

```
Router#show performance-measurement interfaces be12 detail | i Local MAC Address
  Local MAC Address                  : bc8d.xxx.1d04
```