

Configure Performance Measurement

Table 1: Feature History

Feature Name	Release Information	Description
Segment Routing Performance Measurement Delay Measurement Using RFC 5357 (TWAMP Light)	Cisco IOS XE Amsterdam 17.3.1	This feature enables hardware timestamping. The Performance Measurement (PM) for link delay uses the light version of Two-Way Active Measurement Protocol (TWAMP) over IP and UDP defined in Appendix I of RFC 5357. TWAMP provides an alternative for interoperability when RFC 6374 is not used.

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

Starting from Cisco IOS XE Release 17.3.1, hardware timestamping is supported. The time stamps help ensure that the routers achieve outstanding results when deploying IEEE 1588-2008 protocols for frequency and phase synchronization.

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

Table 2: Performance Measurement Functionalities

Functionality	Details
Profiles	Configure different profiles for different types of delay measurements. Delay profile type interfaces are 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.
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.

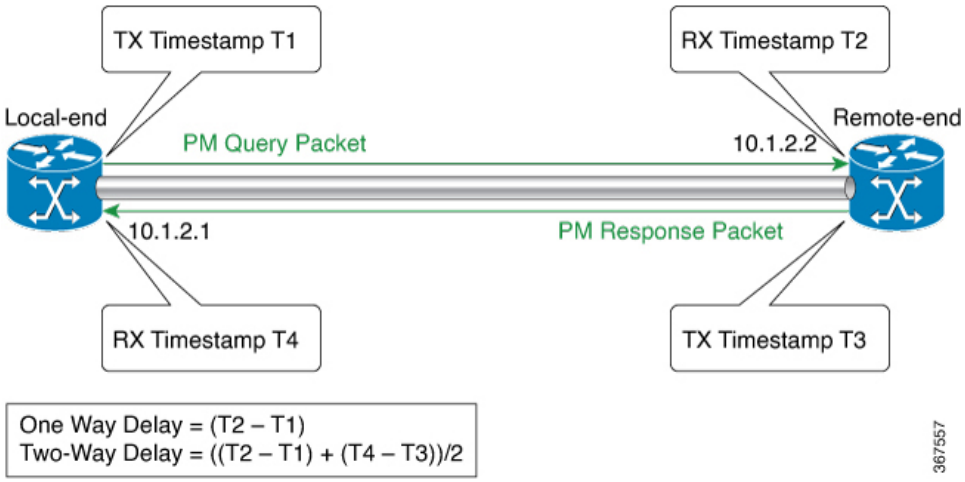
- Link Delay Measurement, on page 2
- End-to-End Delay Measurement, on page 7
- SR-PCE: Enabling SR PM Delay or Liveness for PCE-Initiated Policies, on page 10
- Telemetry (Model-Based Telemetry and Event-Based Telemetry) Support for Performance Measurement, on page 14
- Configuring UDP Destination Port, on page 21

Link Delay Measurement

The PM for link delay uses the light version of Two-Way Active Measurement Protocol (TWAMP) over IP and UDP defined in Appendix I of RFC 5357. Hence, only TWAMP test sessions are implemented and not the TWAMP control protocol. TWAMP provides an alternative for interoperability when RFC 6374 is not used. TWAMP packets are carried over IP and UDP. Thus, the dependency on MPLS dataplane is eliminated.

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

Figure 1: 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 broadcast links, only point-to-point (P2P) links are supported. P2P configuration on IGP is required for flooding the value.
- Only TWAMP protocol based PM probes are supported. MPLS-GAL based PM probes are not supported.
- For one-way delay measurement, clocks should be synchronized on two end-point nodes of the link using PTP.
- Two-way performance measurements on the RSP3 module is supported with the **sr_5_label_push_enable** or **sr_pfp_enable** SDM template only.
- The **control-word** command used in pseudowire configuration is not supported with the **sr_5_label_push_enable** SDM template.

PM Link Delay: Default Values for Different Parameters

The default values for the different parameters in the PM for link delay is given as follows:

- probe: The default 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).
- interval: The default probe interval is 30 seconds. The range is from 30 to 3600 seconds.
- burst count: The default value is 10 and range is from 1 to 30.
- burst interval: The default value is 3000 milliseconds and the range is from 30 to 15000 milliseconds.
- 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: The default value is 1000 microseconds (usec) and the range is from 0 to 100000 microseconds.
- accelerated advertisement: Accelerated advertisement is disabled by default.
- accelerated-advertisement threshold: The default value is 20 percent and the range is from 0 to 100 percent.
- accelerated-advertisement minimum: The default value is 1000 microseconds and the range is from 1 to 100000 microseconds.

Configuration Example: PM for Link Delay

This example shows how to configure performance-measurement functionalities for link delay as a global default profile.

```
R1(config)#performance-measurement
R1(config-perf-meas)#delay-profile interfaces
R1(config-pm-dm-intf)#probe
R1(config-pm-dm-intf-probe)#interval 40
R1(config-pm-dm-intf-probe)#protocol twamp-light
R1(config-pm-dm-intf-probe)#burst count 5
R1(config-pm-dm-intf-probe-burst)#interval 40
R1(config-pm-dm-intf-probe-burst)#exit
R1(config-pm-dm-intf-probe)#exit
R1(config-pm-dm-intf)#advertisement periodic
R1(config-pm-dm-intf-adv-per)#interval 100
R1(config-pm-dm-intf-adv-per)#threshold 80
R1(config-pm-dm-intf-adv-per)#minimum-change 5000
R1(config-pm-dm-intf-adv-per)#exit
R1(config-pm-dm-intf)#advertisement accelerated
R1(config-pm-dm-intf-adv-acc)#threshold 30
R1(config-pm-dm-intf-adv-acc)#minimum-change 1100
R1(config-pm-dm-intf-adv-acc)#exit
```

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

```
R1(config)#performance-measurement
R1(config-perf-meas)#interface gigabitEthernet0/3/3
R1(config-pm-intf)#delay-measurement
```

```
R1(config-pm-intf-dm)#next-hop ipv4 170.50.62.1
R1(config-pm-intf)#exit
```

Verification: PM Link Delay Configuration

This example shows how to use the **show performance-measurement summary** [**detail**] command to verify the PM for link-delay configuration.

```
R1#show performance-measurement summary detail
Total interfaces                                : 3
Maximum PPS                                     : 100 pkts/sec

Interface Delay-Measurement:
  Total sessions                                : 3
  Profile configuration:
    Measurement Type                            : Two-Way
    Computation interval                        : 30 seconds
    Burst interval                              : 3000 mSec
    Burst count                                 : 10 packets
    Protocol                                    : TWAMP-Lite Unauth
    HW Timestamp Supported                      : No
    Periodic advertisement                      : Enabled
      Interval                                  : 30 (effective: 30) sec
      Threshold                                 : 100%
      Minimum-Change                            : 100000 uSec
    Accelerated advertisement                   : Enabled
      Threshold                                 : 100%
      Minimum-Change                            : 100000 uSec
    Threshold crossing check                    : Minimum-delay
  Counters:
    Packets:
      Total sent                                : 293020
      Total received                            : 293016
    Errors:
        TX:
          Total interface down                  : 0
          Total no MPLS caps                    : 0
          Total no IP address                   : 0
          Total other                           : 19
        RX:
          Total negative delay                  : 144
          Total delay threshold exceeded        : 0
          Total missing TX timestamp            : 0
          Total missing RX timestamp            : 0
          Total probe full                      : 0
          Total probe not started               : 0
          Total control code error              : 0
          Total control code notif              : 0
    Probes:
      Total started                             : 29306
      Total completed                           : 29155
      Total incomplete                          : 148
      Total advertisements                      : 3

Global Delay Counters:
  Total packets sent                            : 293020
  Total query packets received                  : 293016
  Total invalid session id                      : 0
  Total no session                              : 0

HW Support for MPLS-GAL [RFC6374] timestamp   : No
HW Support for TWAMP [RF5357] timestamp       : No
```

```
HW Support for 64 bit timestamp              : No
HW Support for IPv4 UDP Cheksum              : No
```

This example shows how to use the **show performance-measurement interfaces** [*interface-name*] [**detail**] command to verify the PM for link-delay configuration.

```
R1#show performance-measurement interfaces detail
Interface Name: GigabitEthernet0/2/3 (ifh: 0xA)
  Delay-Measurement           : Enabled
  Local IPV4 Address          : 170.50.62.2
  Local IPV6 Address          : ::
  State                       : Up

  Delay Measurement session:
    Session ID                : 1

    Last advertisement:
      Advertised at: 09:21:08  12 2019 (439879 seconds ago)
      Advertised reason: Advertise delay config
      Advertised delays (uSec): avg: 2000, min: 2000, max: 2000, variance: 0

    Next advertisement:
      Check scheduled at the end of the current probe (roughly every 30 seconds)
      No probes completed
      Rolling average (uSec): 3146

    Current Probe:
      Started at 11:32:17  17 2019 (10 seconds ago)
      Packets Sent: 4, received: 4
      Measured delays (uSec): avg: 1999, min: 1500, max: 2499, variance: 499
      Probe samples:
              Packet Rx Timestamp Measured Delay
                11:32:17  17 2019 1999999
                11:32:20  17 2019 1500000
                11:32:23  17 2019 2499999
                11:32:26  17 2019 1999999
      Next probe scheduled at 11:32:46  17 2019 (in 19 seconds)
      Next burst packet will be sent in 1 seconds
R1#
```

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 interfaces [*interface*] probe	Displays the PM link-delay probe history for interfaces.
show performance-measurement history interfaces [*interface*] aggr	Displays the PM link-delay aggregated history for interfaces.
show performance-measurement counters [interface *interface*]	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	Displays the PM for link-delay summary on the remote-end router (responder).

Command	Description
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*]	Displays the PM link-delay session counters on the remote-end router.

End-to-End Delay Measurement

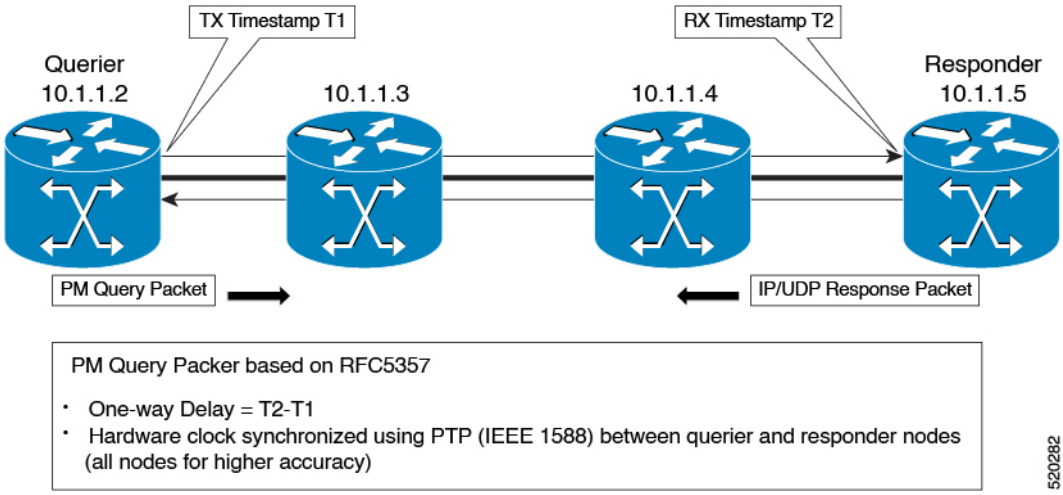
Table 3: Feature History

Feature Name	Release Information	Description
Segment Routing Performance Measurement End-to-End Delay Measurement	Cisco IOS XE Amsterdam 17.3.1	This feature allows to monitor the end-to-end delay experienced by the traffic sent over a Segment Routing policy. This feature ensures the delay does not exceed the specified threshold value and violate the SLAs. Use this feature to apply extended TE link delay metric (minimum delay value) to compute paths for Segment Routing policies as an optimization metric or as an accumulated delay bound.

Starting from Cisco IOS XE Release 17.3.1, end-to-end delay measurement feature is introduced for Segment Routing Performance Management. Use this feature to monitor the end-to-end delay experienced by the traffic sent over a Segment Routing policy. This feature ensures the delay does not exceed the specified threshold value and violate the SLAs. You can verify the end-to-end delay values before activating the candidate-path or the segment-list of the Segment Routing policy in the forwarding table. You can also use the end-to-end delay values to deactivate the active candidate-path or the segment-list of the Segment Routing Policy in the forwarding table. Use this feature to apply extended TE link delay metric (minimum delay value) to compute paths for Segment Routing policies as an optimization metric or as an accumulated delay bound.

The following figure explains the PM query and response for end-to-end delay measurement.

Figure 2: Performance Measurement for End-to-End Delay Measurement



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

1. The querier router sends PM query packets periodically to the responder router once the egress line card on the router applies timestamps on packets.
2. Ingress line card on the responder router applies time-stamps on packets when they are received.
3. The end-to-end delay value of an SR Policy is different than the path computation result (the sum of TE link delay metrics) due to several factors like queuing delay within the routers.

Configuration Example: PM for End-to-End Delay Management

These examples show how to configure on-demand segment routing policy for end-to-end delay management.

```
#show running-config | s on-demand color 800
on-demand color 800 ----------------------------------------------------------> SR ODN Policy
authorize
performance-measurement ------------------------------------------------------> SR PM CLI
delay-measurement ------------------------------------------------------------> SR PM CLI
candidate-paths
preference 1
constraints
segments
dataplane mpls
!
!
dynamic
pcep
metric
type delay
!
!
!
#

#show segment-routing traffic-eng policy name *216.216.216.216|800

Name: *216.216.216.216|800 (Color: 800 End-point: 216.216.216.216)
Owners : BGP
Status:
```

```
Admin: up, Operational: up for 01:27:24 (since 11-29 04:41:36.053)
Candidate-paths:
Preference 1 (BGP):
Dynamic (pce 12.12.12.12) (active)
Weight: 0, Metric Type: DELAY
Metric Type: DELAY, Path Accumulated Metric: 330
16011 [Prefix-SID, 205.205.205.205]
1133 [Adjacency-SID, 170.50.72.1 - 170.50.72.2]
16009 [Prefix-SID, 216.216.216.216]
Attributes:
Binding SID: 1218
Allocation mode: dynamic
State: Programmed
IPv6 caps enabled
#
```

This example shows how to configure performance-measurement functionalities for end-to-end delay management as a global default profile.

```
R1(config)#performance-measurement
R1(config-perf-meas)#delay-profile sr-policy
R1(config-pm-dm-intf)#probe
R1(config-pm-dm-intf-probe)#interval 40
R1(config-pm-dm-intf-probe)#protocol twamp-light
R1(config-pm-dm-intf-probe)#burst count 5
R1(config-pm-dm-intf-probe-burst)#interval 40
R1(config-pm-dm-intf-probe-burst)#exit
R1(config-pm-dm-intf-probe)#exit
R1(config-pm-dm-intf)#advertisement periodic
R1(config-pm-dm-intf-adv-per)#interval 100
R1(config-pm-dm-intf-adv-per)#threshold 80
R1(config-pm-dm-intf-adv-per)#minimum-change 5000
R1(config-pm-dm-intf-adv-per)#exit
R1(config-pm-dm-intf)#advertisement accelerated
R1(config-pm-dm-intf-adv-acc)#threshold 30
R1(config-pm-dm-intf-adv-acc)#minimum-change 1100
R1(config-pm-dm-intf-adv-acc)#exit
```

This example shows how to enable PM for end-to-end delay management over an interface.

```
R1(config)#performance-measurement
R1(config-perf-meas)#interface gigabitEthernet0/3/3
R1(config-pm-intf)#delay-measurement
R1(config-pm-intf-dm)#next-hop ipv4 170.50.62.1
R1(config-pm-intf)#exit
```

Verification: PM End-to-End Delay Management Configuration

This example shows how to use the **show performance-measurement summary** command to verify the PM for end-to-end delay management configuration.

```
R1#show performance-measurement summary
Total interfaces                              : 6
Total SR Policies                             : 1
Maximum PPS                                   : 1000 pkts/sec

SR Policy Delay-Measurement:
  Total sessions                              : 1
  Profile configuration:
    Measurement Type                          : One-Way
```

```
    Computation interval                      : 30 seconds
    Burst interval                            : 3000 mSec
    Burst count                               : 10
    Protocol                                  : TWAMP-Lite Unauth
    HW Timestamp Supported                    : Yes
    Periodic advertisement                    : Enabled
      Interval                                : 30 (effective: 30) sec
      Threshold                               : 15%
      Minimum-Change                          : 600 uSec
    Accelerated advertisement                 : Enabled
      Threshold                               : 25%
      Minimum-Change                          : 900 uSec
    Threshold crossing check                  : Minimum-delay
  Counters:
    Packets:
      Total sent                              : 334
      Total received                          : 0
    Errors:
      Total sent errors                       : 0
      Total received errors                   : 0
    Probes:
      Total started                           : 33
      Total completed                         : 0
      Total incomplete                        : 33
      Total advertisements                    : 0

Global Delay Counters:
  Total packets sent                          : 1251
  Total query packets received                : 917
  Total invalid session id                    : 0
  Total no session                            : 0

HW Support for MPLS-GAL [RFC6374] timestamp   : No
HW Support for TWAMP [RF5357] timestamp       : Yes
HW Support for 64 bit timestamp               : Yes
HW Support for IPv4 UDP Cheksum               : No
R1#
```

SR-PCE: Enabling SR PM Delay or Liveness for PCE-Initiated Policies

Table 4: Feature History

Feature Name	Release Information	Description
SR-PCE: Enabling SR PM Delay or Liveness for PCE-Initiated Policies	Cisco IOS XE Bengaluru 17.6.1	This feature enables the Path Computation Element (PCE) that can provision a Segment Routing Traffic Engineering (SR-TE) policy to mitigate link congestion. Prior to this release, you could only enable PM link and delay measurement using CLI-based policies. Starting with this release, you can also use PCE to enable PM link and delay measurement.

The Path Computation Element (PCE) can provision a Segment Routing Traffic Engineering (SR-TE) policy to mitigate link congestion. The Path Computation Element Protocol (PCEP) describes a set of procedures by which a path computation client (PCC) can report and delegate control of head-end label switched paths (LSPs) sourced from the PCC to a PCE peer. The PCE can request the PCC to update and modify parameters of LSPs it controls. The stateful model also enables a PCC to allow the PCE to initiate computations allowing the PCE to perform network-wide orchestration. SR-PCE learns topology information by way of IGP (OSPF or IS-IS) or through BGP Link-State (BGP-LS).

Prior to Cisco IOS XE Bengaluru Release 17.6.1, you could only enable PM link and delay measurement using CLI-based policies. Starting with Cisco IOS XE Bengaluru Release 17.6.1, you can also use PCE to enable PM link and delay measurement.

Autoroute announcement is a steering mechanism in which IGPs automatically use the policy for destination's downstream of the policy end point. Autoroute announcement is performed using Cisco Crossworks Optimization Engine (COE). COE provides real-time network optimization allowing operators to maximize network utilization effectively and increase service velocity.

A PCE collects various pieces of network information to determine traffic flows causing link congestion. The PCE computes a suitable path to divert those flows and to alleviate the congestion. The PCE then deploys the SR-TE policy to divert the traffic leading to the congestion using the Stateful Path Computation Element Protocol (PCEP) to provision the policy. When the congestion is alleviated, the SR-TE policy is removed.

The PCEP message contains SID list to be deployed by the head-end. Path Computation Client (PCC) profiles allow activation of autoroute announce for the policy provisioned by PCEP, using the profile IDs. The profile ID on the PCE and PCC should match, otherwise the policy is not provisioned. For example, if the PCE provisions a policy with profile ID 1 and the head-end where the policy is being provisioned also has the PCC profile ID 1 configured with autoroute announce, COE-PCE initiated SR policy is activated for that policy.

SR-PCE is capable of computing paths using the following methods:

- TE metric—SR-PCE uses the TE metric in its path calculations to optimize cumulative TE metric.
- IGP metric—SR-PCE uses the IGP metric in its path calculations to optimize reachability.
- LSP Disjointness—SR-PCE uses the path computation algorithms to compute a pair of disjoint LSPs. The disjoint paths can originate from the same head-end or different head-ends. Disjoint level refers to the type of resources that should not be shared by the two computed paths. SR-PCE supports the following disjoint path computations:
  - Link – Specifies that links are not shared on the computed paths.
  - Node – Specifies that nodes are not shared on the computed paths.
  - SRLG – Specifies that links with the same SRLG value are not shared on the computed paths.
  - SRLG-node – Specifies that SRLG and nodes are not shared on the computed paths.

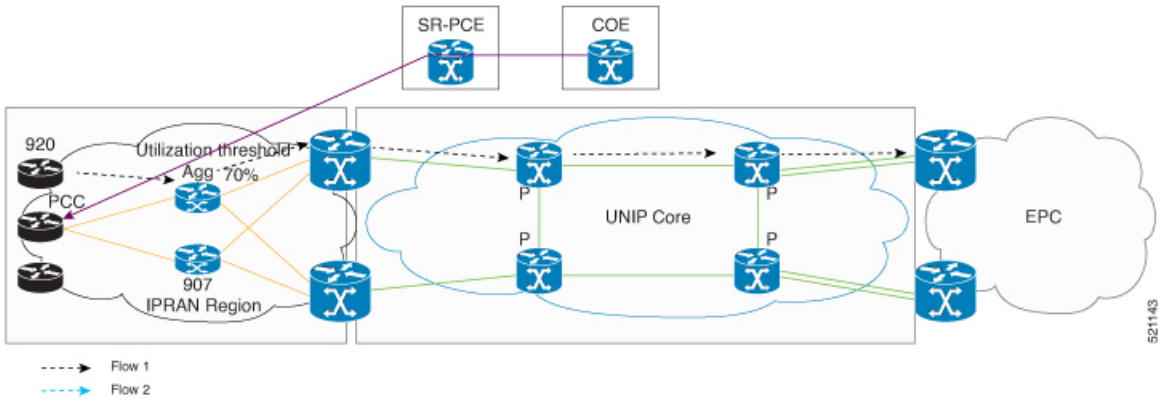
COE-PCE Initiated SR Policy

The following topology shows how an SR-PCE policy is initiated from COE:

- SR policy is configured on the COE with profile ID.
- COE pushes the SR policy to PCE and PCE forwards the SR policy to PCC.
- Profile ID on PCC is matched with the profile ID on COE-PCE.

- OSPF autoroute announce is configured on the PCC.
- The policy gets provisioned.
- The data traffic now adheres to the SR policy that is pushed from the COE.
- Complete SR Policy manipulation occurs only on COE.

Figure 3: COE-PCE Initiated SR Policy



Configure SR-PCE: Enabling SR-PM Delay or Liveness for PCE-Initiated Policies

To enable SR-PM delay or liveness for PCE-Initiated policies, configure PCC and PCE nodes.

Configure PCC Node:

To configure PCC node:

```
pcc
  pce address 9.9.9.9 source-address 10.0.0.1
  report-all
  profile 1
   autoroute
    include all
   !
   performance-measurement
    delay-measurement
     profile test
     liveness-detection
      invalidation-action down
   !
  !
```

Configure PCE Node:

To configure PCE node:

```
pce
address ipv4 9.9.9.9
api
!
peer ipv4 10.0.0.1
!
peer ipv4 2.2.2.2
!
peer ipv4 4.4.4.4
```

```
!
segment-routing
  traffic-eng
    segment-list name srte11
    index 1 mpls adjacency 11.11.11.2
    index 2 mpls adjacency 13.13.13.2
    index 3 mpls adjacency 17.17.17.2
   !
   segment-list name srte12
    index 1 mpls adjacency 12.12.12.2
    index 2 mpls adjacency 15.15.15.2
    index 3 mpls adjacency 18.18.18.2
   !
   segment-list name srte13
    index 1 mpls adjacency 21.21.21.2
    index 2 mpls adjacency 22.22.22.2
    index 3 mpls adjacency 23.23.23.2
   !

   peer ipv4 10.0.0.1
    policy test
     color 10 end-point ipv4 2.2.2.2
     candidate-paths
      preference 100
       explicit segment-list srte11
       !
      !
      preference 200
       explicit segment-list srte13
       !
      !
      preference 300
       explicit segment-list srte12
       !
      !
      preference 400
       explicit segment-list srte11
       !
      !
     !
     profile-id 1
    !
```

Verification of SR-PCE: Enabling SR-PM Delay or Liveness for PCE-Initiated Policies

Use the **show segment-routing traffic-engineering policy all** command to verify the SR-PM delay or liveness for PCE-initiated policies configuration.

```
PE1(config)#do show segment-routing traffic-engineering policy all
Name: *2.2.2.2|10 (Color: 10 End-point: 2.2.2.2)
  Owners : PCEP
  Status:
    Admin: up, Operational: up for 13:50:38 (since 04-27 20:27:25.138)
  Candidate-paths:
    Preference 400 (PCEP):
      PM State: Up
      PCC profile: 1
      Dynamic (pce 9.9.9.9) (active)
        Metric Type: TE, Path Accumulated Metric: 0
          37 [Adjacency-SID, 11.11.11.1 - 11.11.11.2]
```

```
        28 [Adjacency-SID, 13.13.13.1 - 13.13.13.2]
        48 [Adjacency-SID, 17.17.17.1 - 17.17.17.2]
Attributes:
  Binding SID: 153
    Allocation mode: dynamic
    State: Programmed
  Autoroute:
    Include all (Strict)
```

Telemetry (Model-Based Telemetry and Event-Based Telemetry) Support for Performance Measurement

Table 5: Feature History

Feature Name	Release Information	Description
Telemetry (Model-Based Telemetry and Event-Based Telemetry) Support for Performance Measurement	Cisco IOS XE Amsterdam 17.3.1	This feature enables Model-Based Telemetry (MDT) and Event-Based Telemetry (EDT) that allow the data to be directed to a configured receiver. This data can be used for analysis and troubleshooting purposes to maintain the health of the network. This feature is supported on Cisco ASR 900 RSP3 module. The **sr_5_label_push_enable** SDM template is mandatory for this feature to function.

Table 6: Feature History

Feature Name	Release Information	Description
Telemetry (Model-Based Telemetry and Event-Based Telemetry) Support for Performance Measurement	Cisco IOS XE Bengaluru 17.4.1	This feature enables Model-Based Telemetry (MDT) and Event-Based Telemetry (EDT) that allow the data to be directed to a configured receiver. This data can be used for analysis and troubleshooting purposes to maintain the health of the network. The **sr_5_label_push_enable** SDM template is mandatory for this feature to function.

Telemetry is the process of measuring the state of the components in a system and transmitting it to a remote location for further processing and analysis.

The demand for data regarding network state, whether to detect hot spots in the network, or to aid decision making on workload placement requires data at a cadence that traditional methods cannot deliver. SNMP, CLI, and Syslog have limitations that restrict automation and scale.

Streaming telemetry lets users direct data to a configured receiver. This data can be used for analysis and troubleshooting purposes to maintain the health of the network. This is achieved by leveraging the capabilities of machine-to-machine communication.

Model-Driven Telemetry (MDT) is an approach for network monitoring in which data is streamed from the network devices continuously using a push model and provides near real-time access to operational statistics. Applications can subscribe to specific data items they need, by using standard-based YANG data models over NETCONF-YANG.

Note The **`sr_5_label_push_enable`** SDM template is mandatory for this feature to function.

Probe, Aggregation, and Advertisement

Probe is a packet sent over a regular interval (probe interval) that carries the information about measurement (for example, delay, loss, and so on). The two types of probes are query and responder.

Aggregation is the process of aggregating the measurement values of the number of probes. The aggregation process is performed at a regular interval of time. The aggregation interval is usually a multiple of theprobe interval; however, it can be as less as a probe interval.

Advertisement is a process of advertising the aggregated values when the measurement values cross the pre-determined threshold values. The advertisement check is performed after every aggregation interval. When the accelerated advertisement is configured, the check is performed in every probe interval.

Configuration Methods of MDT

- **Cadence-Based Telemetry**: Cadence-based Telemetry (CDT) continuously streams data (operational statistics and state transitions) at a configured cadence. The streamed data helps users closely identify patterns in the networks (for example, streaming data about interface counters, and so on). Configuring the interval to any nonzero value sets the subscription for cadence-based telemetry.

  It supports the Histogram Data. Histograms are more complex data type requiring the most processing on a device. Histograms store the frequency of occurrence over a time period and typically use ranges to group similar values. Histogram data provides the following information:

  - Data of the history of the probe, aggregation, and advertisement
  - Data for the last probe, last aggregation, and the last advertisement

- **Event-Based Telemetry:** Event-driven Telemetry (EDT) optimizes data collected at the receiver by streaming data only when a state transition occurs (for example, stream data only when an interface state transitions, IP route updates, and so on). Configuring the sample interval value to zero sets the subscription for event-based telemetry. EDT provides the following information:

  - Delay metrics computed in the last probe-interval (Event: probe-completed)
  - Delay metrics computed in the last aggregation-interval or the end of the periodic advertisement-interval (Event: advertisement-interval expired)

- Delay metrics last flooded in the network (Event: flooding-triggered)

The table below shows the data supported for link delay and end-to-end delay measurement in Oper-model. Oper-model is one of the categories in YANG model testing, where the operation data is pulled from the node.

Performance Measurement	Data Supported
Link Delay Measurement	Interface-last-probes
	interface-last-aggregations
	interface-last-advertisements
	interface-probe-histories
	interface-aggregated-histories
	interface-advertisement-histories
End-to-End Delay Measurement	sr-policy-last-probes
	sr-policy-last-aggregations
	sr-policy-last-advertisements
	sr-policy-probe-histories
	sr-policy-aggregated-histories
	sr-policy-advertisement-histories

For more information on the Telemetry feature, see the Programmability Configuration Guide, Cisco IOS XE Amsterdam 17.1.x.

Configuration Example: Telemetry for Performance Measurement

The following example shows the configuration example of telemetry for performance measurement (End-to-End Delay measurement) for the interface last advertisement option:

```
configure terminal
telemetry ietf subscription 100
encoding encode-kvgpb
filter xpath /performance-measurement/if-delay/last-advertisement
source-address <management-ip-address>
source-vrf management-interface
stream yang-push
update-policy periodic 100
receiver ip address <x.x.x.x> 57344 protocol grpc-tcp
```

The following example shows the sample output of telemetry configuration:

```
Node : <Router>
Subscription : 100
Path :
Cisco-IOS-XE-performance-measurement-oper:performance-measurement/if-delay/last-advertisement

Key : /if-name : GigabitEthernet0/0/13
```

```
/values/avg : 130
/values/min : 106
/values/max : 197
/values/var : 24
/timestamp : 2020-07-28T09:32:44+00:00
/advertised-reason : per-threshold-min
```

The options to configure telemetry performance measurement (Link Delay measurement) are:

- if-name /performance-measurement/if-delay/if-name
- probe is valid /performance-measurement/if-delay/probe-is-valid
- aggr is valid /performance-measurement/if-delay/aggr-is-valid
- adv is valid /performance-measurement/if-delay/adv-is-valid
- last probe /performance-measurement/if-delay/last-probe
- last aggr /performance-measurement/if-delay/last-aggr
- last adv /performance-measurement/if-delay/last-adv
- probe history /performance-measurement/if-delay/probe-history
- aggr-history /performance-measurement/if-delay/aggr-history
- adv history /performance-measurement/if-delay/adv-history

The options to configure telemetry performance measurement (End-to-End Delay measurement) are:

- sr-policy name /performance-measurement/sr-policy-delay/sr-policy-name
- sr-policy probe is valid /performance-measurement/sr-policy-delay/probe-is-valid
- aggr-is-vaid /performance-measurement/sr-policy-delay/aggr-is-valid
- adv-is-valid /performance-measurement/sr-policy-delay/adv-is-valid
- lastprobe /performance-measurement/sr-policy-delay/last-probe
- probe history /performance-measurement/sr-policy-delay/probe-history
- last-aggr /performance-measurement/sr-policy-delay/last-aggr
- aggr-history /performance-measurement/sr-policy-delay/aggr-history
- last-adv /performance-measurement/sr-policy-delay/last-adv
- adv history /performance-measurement/sr-policy-delay/adv-history

Verification of MDT and EDT Support for Performance Measurement

Use the following commands to verify the configuration of MDT and EDT for performance measurement.

```
Router#show performance-measurement history sr-policy probe
SR Policy name: foo
  Candidate-Path:
    Preference               : 10
    Protocol-origin          : Configured
    Discriminator            : 0
    Active                   : Yes
```

```
      Probe Start Timestamp Pkt(TX/RX) Average   Min       Max
          09:59:35  12 2020 3/3        303000    303000    303000
          09:59:30  12 2020 3/3        303000    303000    303000
          09:59:25  12 2020 3/3        302333    302000    303000
          09:59:20  12 2020 3/3        303000    303000    303000
          09:59:15  12 2020 3/3        303000    303000    303000
          09:59:10  12 2020 3/3        303000    303000    303000
          09:59:05  12 2020 3/3        302333    302000    303000
          09:59:00  12 2020 3/3        302333    302000    303000
          09:58:55  12 2020 3/3        303333    303000    304000
          09:58:50  12 2020 3/3        303000    303000    303000
          09:58:45  12 2020 3/3        302000    302000    302000
Segment-list:
    Name                  : SegmentList0
        Probe Start Timestamp Pkt(TX/RX) Average   Min       Max
            09:59:35  12 2020 3/3        303000    303000    303000
            09:59:30  12 2020 3/3        303000    303000    303000
            09:59:25  12 2020 3/3        302333    302000    303000
            09:59:20  12 2020 3/3        303000    303000    303000
            09:59:15  12 2020 3/3        303000    303000    303000
            09:59:10  12 2020 3/3        303000    303000    303000
            09:59:05  12 2020 3/3        302333    302000    303000
            09:59:00  12 2020 3/3        302333    302000    303000
            09:58:55  12 2020 3/3        303333    303000    304000
            09:58:50  12 2020 3/3        303000    303000    303000
            09:58:45  12 2020 3/3        302000    302000    302000
    Atomic path:
      Hops                : 192.168.0.2, 192.168.0.9
      Labels              : 16151
      Outgoing Interface  : Ethernet0/1
      Next Hop            : 11.11.11.2
      Destination         : 192.168.0.9
      Session ID          : 4
          Probe Start Timestamp Pkt(TX/RX) Average   Min       Max
              09:59:35  12 2020 1/1        303000    303000    303000
              09:59:30  12 2020 1/1        303000    303000    303000
              09:59:25  12 2020 1/1        303000    303000    303000
              09:59:20  12 2020 1/1        303000    303000    303000
              09:59:15  12 2020 1/1        303000    303000    303000
              09:59:10  12 2020 1/1        303000    303000    303000
              09:59:05  12 2020 1/1        303000    303000    303000
              09:59:00  12 2020 1/1        303000    303000    303000
              09:58:55  12 2020 1/1        304000    304000    304000
              09:58:50  12 2020 1/1        303000    303000    303000
              09:58:45  12 2020 1/1        302000    302000    302000
    Atomic path:
      Hops                : 192.168.0.2, 192.168.0.9
      Labels              : 16151
      Outgoing Interface  : Ethernet0/2
      Next Hop            : 12.12.12.2
      Destination         : 192.168.0.9
      Session ID          : 5
          Probe Start Timestamp Pkt(TX/RX) Average   Min       Max
              09:59:35  12 2020 1/1        303000    303000    303000
              09:59:30  12 2020 1/1        303000    303000    303000
              09:59:25  12 2020 1/1        302000    302000    302000
              09:59:20  12 2020 1/1        303000    303000    303000
              09:59:15  12 2020 1/1        303000    303000    303000
              09:59:10  12 2020 1/1        303000    303000    303000
              09:59:05  12 2020 1/1        302000    302000    302000
              09:59:00  12 2020 1/1        302000    302000    302000
              09:58:55  12 2020 1/1        303000    303000    303000
              09:58:50  12 2020 1/1        303000    303000    303000
              09:58:45  12 2020 1/1        302000    302000    302000
```

```
    Atomic path:
      Hops                  : 192.168.0.2, 192.168.0.9
      Labels                : 16151
      Outgoing Interface    : Ethernet0/3
      Next Hop              : 13.13.13.2
      Destination           : 192.168.0.9
      Session ID            : 6
          Probe Start Timestamp Pkt(TX/RX) Average   Min       Max
              09:59:35  12 2020 1/1        303000    303000    303000
              09:59:30  12 2020 1/1        303000    303000    303000
              09:59:25  12 2020 1/1        302000    302000    302000
              09:59:20  12 2020 1/1        303000    303000    303000
              09:59:15  12 2020 1/1        303000    303000    303000
              09:59:10  12 2020 1/1        303000    303000    303000
              09:59:05  12 2020 1/1        302000    302000    302000
              09:59:00  12 2020 1/1        302000    302000    302000
              09:58:55  12 2020 1/1        303000    303000    303000
              09:58:50  12 2020 1/1        303000    303000    303000
              09:58:45  12 2020 1/1        302000    302000    302000

Router#show performance-measurement history sr-policy aggregation
SR Policy name: foo
  Candidate-Path:
    Preference              : 10
    Protocol-origin         : Configured
    Discriminator           : 0
    Active                  : Yes
        Aggregation Timestamp Average   Min       Max       Action
            09:59:12  12 2020 302666    302000    304000    FIRST
    Segment-list:
      Name                  : SegmentList0
          Aggregation Timestamp Average   Min       Max       Action
              09:59:12  12 2020 302666    302000    304000    FIRST
      Atomic path:
        Hops                : 192.168.0.2, 192.168.0.9
        Labels              : 16151
        Outgoing Interface  : Ethernet0/1
        Next Hop            : 11.11.11.2
        Destination         : 192.168.0.9
        Session ID          : 4
            Aggregation Timestamp Average   Min       Max       Action
                09:59:12  12 2020 303000    302000    304000    FIRST
      Atomic path:
        Hops                : 192.168.0.2, 192.168.0.9
        Labels              : 16151
        Outgoing Interface  : Ethernet0/2
        Next Hop            : 12.12.12.2
        Destination         : 192.168.0.9
        Session ID          : 5
            Aggregation Timestamp Average   Min       Max       Action
                09:59:12  12 2020 302499    302000    303000    FIRST
      Atomic path:
        Hops                : 192.168.0.2, 192.168.0.9
        Labels              : 16151
        Outgoing Interface  : Ethernet0/3
        Next Hop            : 13.13.13.2
        Destination         : 192.168.0.9
        Session ID          : 6
            Aggregation Timestamp Average   Min       Max       Action
                09:59:12  12 2020 302499    302000    303000    FIRST

Router#show performance-measurement history sr-policy advertisement

SR Policy name: foo
  Candidate-Path:
```

```
    Preference                : 10
    Protocol-origin           : Configured
    Discriminator             : 0
    Active                    : Yes
      Advertisement Timestamp Average   Min       Max       Action
            09:59:12  12 2020 302666    302000    304000    FIRST
    Segment-list:
      Name                    : SegmentList0
        Advertisement Timestamp Average   Min       Max       Action
              09:59:12  12 2020 302666    302000    304000    FIRST
      Atomic path:
        Hops                  : 192.168.0.2, 192.168.0.9
        Labels                : 16151
        Outgoing Interface    : Ethernet0/1
        Next Hop              : 11.11.11.2
        Destination           : 192.168.0.9
        Session ID            : 4
          Advertisement Timestamp Average   Min       Max       Action
                09:59:12  12 2020 303000    302000    304000    FIRST
      Atomic path:
        Hops                  : 192.168.0.2, 192.168.0.9
        Labels                : 16151
        Outgoing Interface    : Ethernet0/2
        Next Hop              : 12.12.12.2
        Destination           : 192.168.0.9
        Session ID            : 5
          Advertisement Timestamp Average   Min       Max       Action
                09:59:12  12 2020 302499    302000    303000    FIRST
      Atomic path:
        Hops                  : 192.168.0.2, 192.168.0.9
        Labels                : 16151
       Outgoing Interface    : Ethernet0/3
        Next Hop              : 13.13.13.2
        Destination           : 192.168.0.9
        Session ID            : 6
          Advertisement Timestamp Average   Min       Max       Action
                09:59:12  12 2020 302499    302000    303000    FIRST

Router#show performance-measurement history interfaces advertisement
Interface Name: Ethernet0/1 (ifh: 0x3)
  Delay-Measurement history (uSec):
      Advertisement Timestamp Average   Min       Max       Action
            10:10:41  12 2020 204600    1         329999    FIRST

Router#show performance-measurement history interfaces aggregation
Interface Name: Ethernet0/1 (ifh: 0x3)
  Delay-Measurement history (uSec):
        Aggregation Timestamp Average   Min       Max       Action
            10:10:41  12 2020 189405    1         329999    FIRST

Router#show performance-measurement history interfaces probe
Interface Name: Ethernet0/1 (ifh: 0x3)
  Delay-Measurement history (uSec):
        Probe Start Timestamp Pkt(TX/RX) Average   Min       Max
            10:10:45  12 2020 3/3        202666    202499    202999
            10:10:35  12 2020 3/3        202999    202999    202999
            10:10:25  12 2020 3/3        202999    202999    202999
            10:10:15  12 2020 3/3        202333    201999    202500
            10:10:05  12 2020 3/3        203166    202999    203499
            10:09:55  12 2020 3/3        202999    202999    202999
            10:09:45  12 2020 3/3        249999    209999    329999
            10:09:35  12 2020 3/3        319999    309999    329999
            10:09:24  12 2020 3/3        326666    319999    329999
            10:09:14  12 2020 3/3        171499    499       299999
```

```
            10:09:04  12 2020 3/3        499       499       499
            10:08:54  12 2020 3/3        333       1         499
```

Use the command below to verify telemetry for End-to-End Delay Measuement:

```
Router#show performance-measurement history interfaces name gigabitEthernet <> advertisement
Interface Name: <> (ifh: 0x14)
  Delay-Measurement history (uSec):
      Advertisement Timestamp Average   Min       Max       Action
            09:56:00  21 2020 161       100       462       PER-MIN
```

Configuring UDP Destination Port

When you specify PM-UDP protocol, you need to configure the 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. The UDP port for each PM measurement probe type must match on querier and responder nodes.

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

Note Starting with Cisco IOS XE Amsterdam 17.3.1 release, the default values for UDP destination ports are available; hence, it is not mandatory to configure the UDP destination ports.

This example shows how to configure the UDP destination port.

```
R1(config)#performance-measurement
R1(config-perf-meas)#protocol twamp-light
R1(config-pm-twamp)#measurement delay
R1(config-pm-twamp-delay)#unauthenticated
R1(config-pm-twamp-delay-unauth)#querier-dst-port 11222 querier-src-port 11333
R1(config-pm-twamp-delay-unauth)#exit
```