Configuring Ethernet OAM

This module describes the configuration of Ethernet Operations, Administration, and Maintenance (OAM):

Table 1: Feature Information Table

Release	Modification
Release 7.3.1	Support for Ethernet Link OAM was introduced.

Information About Configuring Ethernet OAM

To configure Ethernet OAM, you should understand the following concepts:

Ethernet Link OAM

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
Ethernet Link OAM	Release 7.3.1	This feature allows service providers to monitor the quality of the connections on a MAN or WAN. Service providers can monitor specific events, and take actions on events. Ethernet link OAM operates on a single, physical link and it can be configured to monitor either side or both sides of that link.

Ethernet as a Metro Area Network (MAN) or a Wide Area Network (WAN) technology benefits greatly from the implementation of Operations, Administration and Maintenance (OAM) features. Ethernet link OAM features allow Service Providers to monitor the quality of the connections on a MAN or WAN. Service providers can monitor specific events, and take actions on events. Ethernet link OAM operates on a single, physical link and it can be configured to monitor either side or both sides of that link.

Ethernet link OAM can be configured in the following ways:

- A Link OAM profile can be configured, and this profile can be used to set the parameters for multiple interfaces.
- Link OAM can be configured directly on an interface.

  When an interface is also using a link OAM profile, specific parameters that are set in the profile can be overridden by configuring a different value directly on the interface.

An Ethernet Link OAM profile simplifies the process of configuring EOAM features on multiple interfaces. An Ethernet OAM profile, and all of its features, can be referenced by other interfaces, allowing other interfaces to inherit the features of that Ethernet OAM profile.

Individual Ethernet link OAM features can be configured on individual interfaces without being part of a profile. In these cases, the individually configured features always override the features in the profile.

The preferred method of configuring custom EOAM settings is to create an EOAM profile in Ethernet configuration mode and then attach it to an individual interface or to multiple interfaces.

These standard Ethernet Link OAM features are supported on the router:

Neighbor Discovery

Neighbor discovery enables each end of a link to learn the OAM capabilities of the other end and establish an OAM peer relationship. Each end also can require that the peer have certain capabilities before it will establish a session. You can configure certain actions to be taken if there is a capabilities conflict or if a discovery process times out, using the **action capabilities-conflict** or **action discovery-timeout** commands.

EFD

Ethernet Fault Detection (EFD) is a mechanism that allows Ethernet OAM protocols to control the `line protocol` state of an interface.

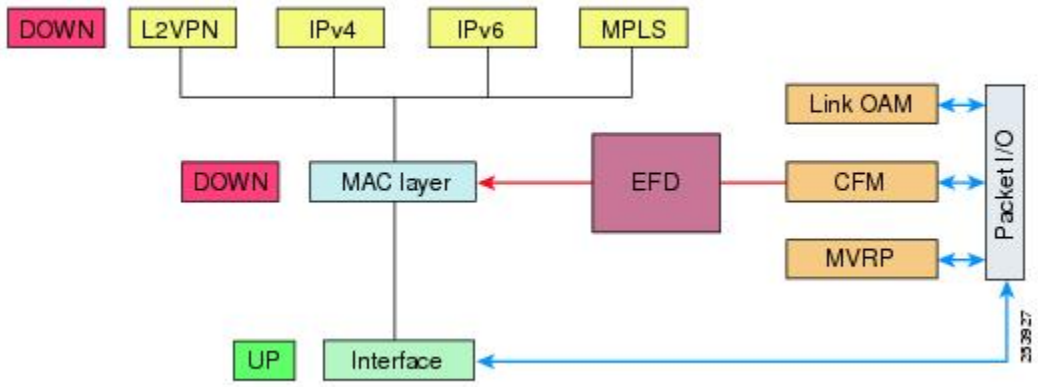
Unlike many other interface types, Ethernet interfaces do not have a line protocol, whose state is independent from that of the interface. For Ethernet interfaces, this role is handled by the physical-layer Ethernet protocol itself, and therefore if the interface is physically up, then it is available and traffic can flow.

EFD changes this to allow EOAM to act as the line protocol for Ethernet interfaces. This allows EOAM to control the interface state so that if a EOAM defect (such as AIS or loss of continuity) is detected with an expected peer MEP, the interface can be shut down. This not only stops traffic flow, but also triggers actions in any higher-level protocols to route around the problem. For example, in the case of Layer 2 interfaces, the MAC table would be cleared and MSTP would reconverge. For Layer 3 interfaces, the ARP cache would be cleared and potentially the IGP would reconverge.

Note EFD can only be used for down MEPs. When EFD is used to shut down the interface, the EOAM frames continue to flow. This allows EOAM to detect when the problem has been resolved, and thus bring the interface backup automatically.

This figure shows EOAM detection of an error on one of its sessions EFD signaling an error to the corresponding MAC layer for the interface. This triggers the MAC to go to a down state, which further triggers all higher level protocols (Layer 2 pseudowires, IP protocols, and so on) to go down and also trigger a reconvergence where possible. As soon as EOAM detects there is no longer any error, it can signal to EFD and all protocols will once again go active.

Figure 1: EOAM Error Detection and EFD Trigger



MIB Retrieval

MIB retrieval enables an OAM peer on one side of an interface to get the MIB variables from the remote side of the link. The MIB variables that are retrieved from the remote OAM peer are READ ONLY.

Miswiring Detection (Cisco-Proprietary)

Miswiring Detection is a Cisco-proprietary feature that uses the 32-bit vendor field in every Information OAMPDU to identify potential miswiring cases.

SNMP Traps

SNMP traps can be enabled or disabled on an Ethernet OAM interface.

Configuration Examples for Ethernet OAM

This section provides the following configuration examples:

Configuring Ethernet OAM Features on an Individual Interface: Example

This example shows how to configure Ethernet OAM features on an individual interface:

```
configure
 interface TenGigE 0/1/0/0
  ethernet oam
  connection timeout 30
  mib-retrieval
  link-monitor
   frame window milliseconds 60000
   frame threshold low 10000000 high 60000000
   frame-period window milliseconds 60000
   frame-period threshold ppm low 100 high 120000
   frame-seconds window milliseconds 900000
   frame-seconds threshold low 3 high 900
   symbol-period window milliseconds 60000
   symbol-period threshold ppm low 1000000 high 1000000
  exit
  require-remote
   mode active
   mib-retrieval
  exit
  action
   critical-event error-disable-interface
   dying-gasp error-disable-interface
   capabilities-conflict error-disable-interface
   wiring-conflict error-disable-interface
   discovery-timeout error-disable-interface
   session-down error-disable-interface
   commit
```

Configuring an Ethernet OAM Profile Globally: Example

This example shows how to configure an Ethernet OAM profile globally:

```
configure
ethernet oam profile Profile_1
 connection timeout 30
 mib-retrieval
 link-monitor
  frame window milliseconds 60000
```

```
  frame threshold low 10000000 high 60000000
  frame-period window milliseconds 60000
  frame-period threshold ppm low 100 high 1000000
  frame-seconds window milliseconds 900000
  frame-seconds threshold low 3 high 900
  symbol-period window milliseconds 60000
  symbol-period threshold ppm low 100000 high 1000000
 exit
 require-remote
  mode active
  mib-retrieval
 exit
 action
  critical-event error-disable-interface
  dying-gasp error-disable-interface
  capabilities-conflict error-disable-interface
  wiring-conflict error-disable-interface
  discovery-timeout error-disable-interface
  session-down error-disable-interface
  commit
```

Configuring Ethernet OAM Features to Override the Profile on an Individual Interface: Example

This example shows the configuration of Ethernet OAM features in a profile followed by an override of that configuration on an interface:

```
configure
 ethernet oam profile Profile_1
  mode passive
  action dying-gasp disable
  action critical-event disable
  action discovery-timeout disable
  action session-up disable
  action session-down disable
  action capabilities-conflict disable
  action wiring-conflict disable

    commit

configure
 interface TenGigE 0/1/0/0
  ethernet oam
   profile Profile_1
    mode active
    action dying-gasp log
    action critical-event log
    action discovery-timeout log
    action session-up log
    action session-down log
    action capabilities-conflict log
    action wiring-conflict log

      commit
```

Recovering from error-disable: Example

You can recover an error-disabled interface due to session-down using one of these methods:

- Manually clear the error-disable using the **clear** command.

```
Router# configure
Router(config)# ethernet oam profile Profile_1
Router(config-eoam)# action
Router(config-eoam-action)# clear session-down error-disable-interface
```

- Disable and then re-enable the network link using administrative shutdown commands to reset the connection.

```
Router# configure
Router(config)# interface TenGigE 0/1/0/0
Router(config-if)# shutdown
Router(config-if)# commit
Router(config-if)# no shutdown
Router(config-if)# commit
```

- Configure an auto-recovery timer for this error-disable reason.

```
Router# configure
Router(config)# error-disable recovery cause link-oam-session-down interval 30
Router(config)# commit
```

Clearing Ethernet OAM Statistics on an Interface: Example

This example shows how to clear Ethernet OAM statistics on an interface:

```
RP/0/RP0/CPU0:router# clear ethernet oam statistics interface gigabitethernet 0/1/5/1
```

Enabling SNMP Server Traps on a Router: Example

This example shows how to enable SNMP server traps on a router:

```
configure
 snmp-server traps ethernet oam events
```

Ethernet CFM

Table 3: Feature History Table

Feature name	Release	Description
CFM on bundle member link for connectivity check	Release 7.3.15	This feature introduces support for Connectivity Fault Management (CFM) on bundle members. Earlier, network administrators managed networks by using the fault, configuration, account, performance, security model. CFM is one of a suite of the Ethernet OAM protocols, which uses a combination of keepalive packets and MAC-based pings, and traceroutes to detect faults in a network. With the CFM feature, you: • reduce operating expenses for service operators by reducing network faults and errors • provide end-to-end maintenance of networks
Up MEP and down MEP support in CFM	Release 7.3.15	This feature introduces Maintenance End Points (MEP) entities that you can configure in a domain. MEPs send either CFM frames from the interface where they are configured or CFM frames that are received on other interfaces. MEPs allow you to perform fault management and carry out performance checks.

Ethernet Connectivity Fault Management (CFM) is a service-level OAM protocol that provides tools for monitoring and troubleshooting end-to-end Ethernet services per VLAN. This includes proactive connectivity monitoring, fault verification, and fault isolation. CFM uses standard Ethernet frames and can be run on any physical media that is capable of transporting Ethernet service frames. Unlike most other Ethernet protocols which are restricted to a single physical link, CFM frames can transmit across the entire end-to-end Ethernet network.

CFM is defined in two standards:

- IEEE 802.1ag—Defines the core features of the CFM protocol.

- ITU-T Y.1731—Redefines, but maintains compatibility with the features of IEEE 802.1ag, and defines some additional features.

Ethernet CFM supports these functions of ITU-T Y.1731:

- ETH-CC, ETH-RDI, ETH-LB, ETH-LT—These are equivalent to the corresponding features defined in IEEE 802.1ag.

Note The Linktrace responder procedures defined in IEEE 802.1ag are used rather than the procedures defined in Y.1731; however, these are interoperable.

- ETH-AIS—The reception of ETH-LCK messages is also supported.

Limitations and restrictions

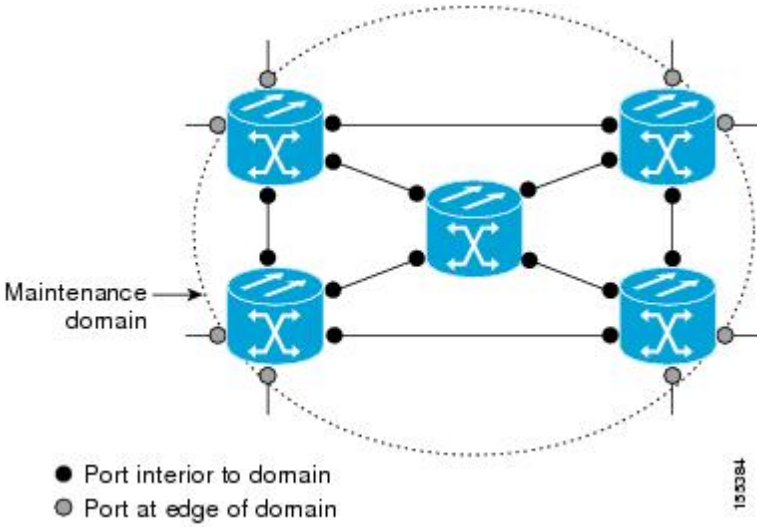
- The system supports only cross-connect.
- MIPs are not supported.
- Supports timer of 1s, 10s, 1m, 10m.
- Supports timer of 100ms, 1s, 10s, 1m, 10m for bundle members.
- L3 interfaces are not supported except for bundle members.
- Down MEPs are only supported for L2 cross-connect and bundle members.
- Multiple MEPs of different directions are not supported on the same interface or Xconnect.
- CFM is not supported on L2 subinterfaces with default encapsulation.
- When configuring CFM down MEP on an interface, ensure that the interface is included in an L2VPN.

Maintenance Domains

To understand how the CFM maintenance model works, you need to understand these concepts and features:

A maintenance domain describes a management space for the purpose of managing and administering a network. A domain is owned and operated by a single entity and defined by the set of interfaces internal to it and at its boundary, as shown in this figure.

Figure 2: CFM Maintenance Domain



A maintenance domain is defined by the bridge ports that are provisioned within it. Domains are assigned maintenance levels, in the range of 0 to 7, by the administrator. The level of the domain is useful in defining the hierarchical relationships of multiple domains.

CFM maintenance domains allow different organizations to use CFM in the same network, but independently. For example, consider a service provider who offers a service to a customer, and to provide that service, they use two other operators in segments of the network. In this environment, CFM can be used in the following ways:

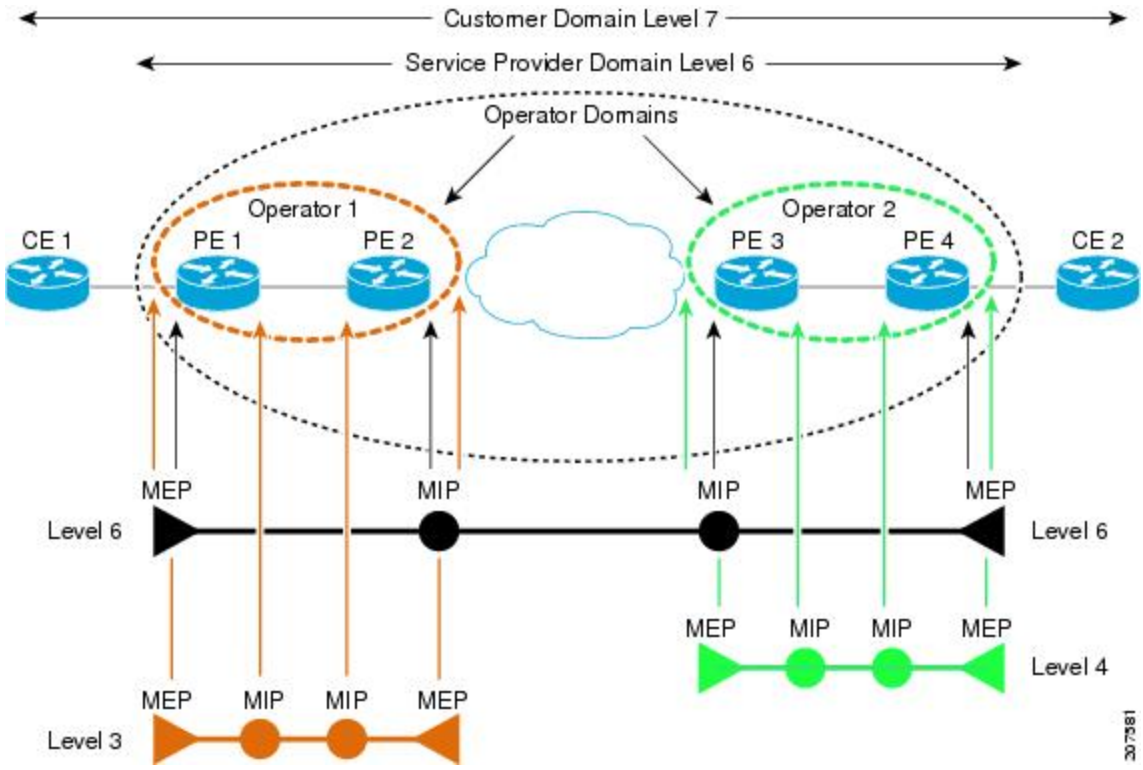
- The customer can use CFM between their CE devices, to verify and manage connectivity across the whole network.
- The service provider can use CFM between their PE devices, to verify and manage the services they are providing.
- Each operator can use CFM within their operator network, to verify and manage connectivity within their network.

Each organization uses a different CFM maintenance domain.

This figure shows an example of the different levels of maintenance domains in a network.

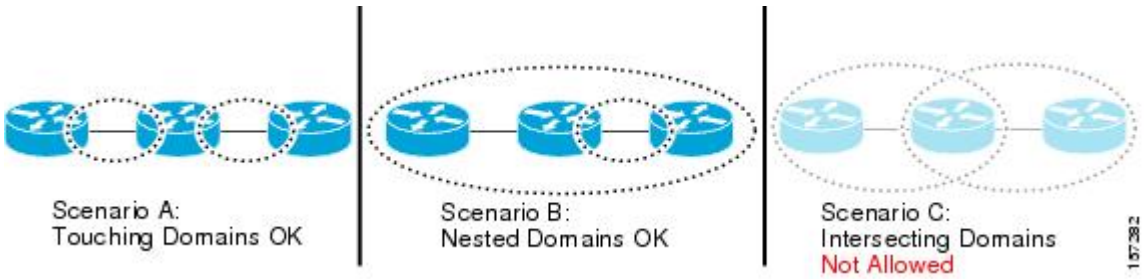
Note In CFM diagrams, the conventions are that triangles represent MEPs, pointing in the direction that the MEP sends CFM frames, and circles represent MIPs.

Figure 3: Different CFM Maintenance Domains Across a Network



To ensure that the CFM frames for each domain do not interfere with each other, each domain is assigned a maintenance level, between 0 and 7. Where domains are nested, as in this example, the encompassing domain must have a higher level than the domain it encloses. In this case, the domain levels must be negotiated between the organizations involved. The maintenance level is carried in all CFM frames that relate to that domain.

CFM maintenance domains may touch or nest, but cannot intersect. This figure illustrates the supported structure for touching and nested domains, and the unsupported intersection of domains.



Services

A CFM service allows an organization to partition its CFM maintenance domain, according to the connectivity within the network. For example, if the network is divided into a number of virtual LANs (VLANs), a CFM service is created for each of these. CFM can then operate independently in each service. It is important that the CFM services match the network topology, so that CFM frames relating to one service cannot be received in a different service. For example, a service provider may use a separate CFM service for each of their customers, to verify and manage connectivity between that customer's end points.

A CFM service is always associated with the maintenance domain that it operates within, and therefore with that domain's maintenance level. All CFM frames relating to the service carry the maintenance level of the corresponding domain.

Note CFM Services are referred to as *Maintenance Associations* in IEEE 802.1ag and as *Maintenance Entity Groups* in ITU-T Y.1731.

Maintenance Points

A CFM Maintenance Point (MP) is an instance of a particular CFM service on a specific interface. CFM only operates on an interface if there is a CFM maintenance point on the interface; otherwise, CFM frames are forwarded transparently through the interface.

A maintenance point is always associated with a particular CFM service, and therefore with a particular maintenance domain at a particular level. Maintenance points generally only process CFM frames at the same level as their associated maintenance domain. Frames at a higher maintenance level are always forwarded transparently, while frames at a lower maintenance level are normally dropped. This helps enforce the maintenance domain hierarchy, and ensures that CFM frames for a particular domain cannot leak out beyond the boundary of the domain.

There are following type(s) of MP(s):

- Maintenance End Points (MEPs)—Created at the edge of the domain. Maintenance end points (MEPs) are members of a particular service within a domain and are responsible for sourcing and sinking CFM frames. They periodically transmit continuity check messages and receive similar messages from other MEPs within their domain. They also transmit traceroute and loopback messages at the request of the administrator. MEPs are responsible for confining CFM messages within the domain.

MEP and CFM Processing Overview

The boundary of a domain is an interface, rather than a bridge or host. Therefore, MEPs can be sub-divided into two categories:

- Down MEPs—Send CFM frames from the interface where they are configured, and process CFM frames received on that interface. Down MEPs transmit AIS messages upward (toward the cross-connect).
- Up MEPs—Send frames into the bridge relay function, as if they had been received on the interface where the MEP is configured. They process CFM frames that have been received on other interfaces, and have been switched through the bridge relay function as if they are going to be sent out of the interface where the MEP is configured. Up MEPs transmit AIS messages downward (toward the wire). However,

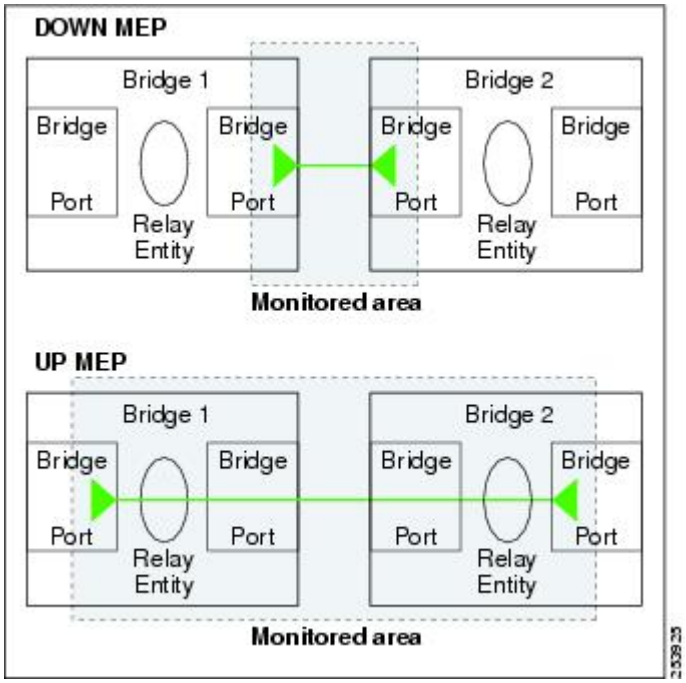
AIS packets are only sent when there is a MIP configured on the same interface as the MEP and at the level of the MIP.

Note

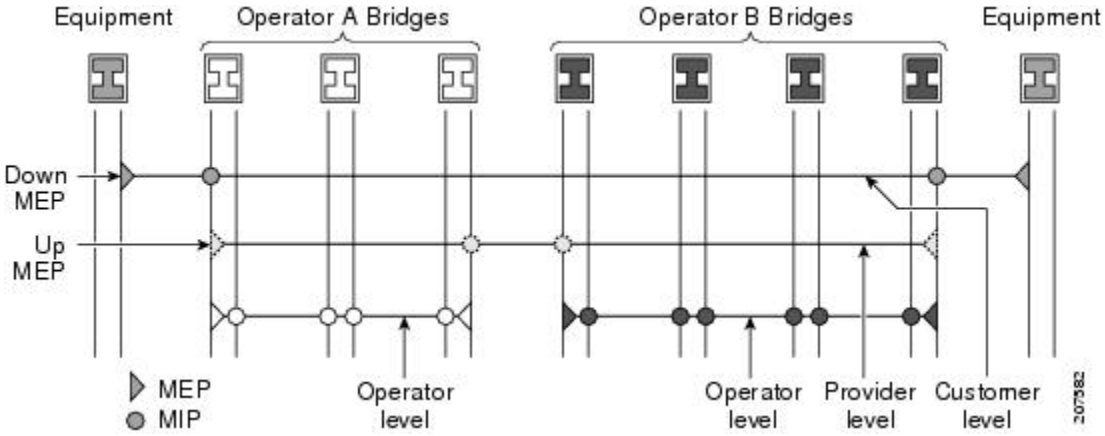
- The terms *Down MEP* and *Up MEP* are defined in the IEEE 802.1ag and ITU-T Y.1731 standards, and refer to the direction that CFM frames are sent from the MEP. The terms should not be confused with the operational status of the MEP.
- The router only supports the "Down MEP level < Up MEP level" configuration.

This figure illustrates the monitored areas for Down and Up MEPs.

Figure 4: Monitored Areas for Down and Up MEPs



This figure shows maintenance points at different levels. Because domains are allowed to nest but not intersect, a MEP at a low level often corresponds with a MEP at a higher level.



Up MEPs can only exist on switched (Layer 2) interfaces, because they send and receive frames from the bridge relay function. Down MEPs can be created on switched (Layer 2) interfaces.

MEPs continue to operate normally if the interface they are created on is blocked by the Spanning Tree Protocol (STP); that is, CFM frames at the level of the MEP continue to be sent and received, according to the direction of the MEP. MEPs never allow CFM frames at the level of the MEP to be forwarded, so the STP block is maintained.

Note A separate set of CFM maintenance levels is created every time a VLAN tag is pushed onto the frame. Therefore, if CFM frames are received on an interface which pushes an additional tag, so as to "tunnel" the frames over part of the network, the CFM frames will not be processed by any MPs within the tunnel, even if they are at the same level. For example, if a CFM MP is created on an interface with an encapsulation that matches a single VLAN tag, any CFM frames that are received at the interface that have two VLAN tags will be forwarded transparently, regardless of the CFM level.

CFM Protocol Messages

The CFM protocol consists of a number of different message types, with different purposes. All CFM messages use the CFM EtherType, and carry the CFM maintenance level for the domain to which they apply.

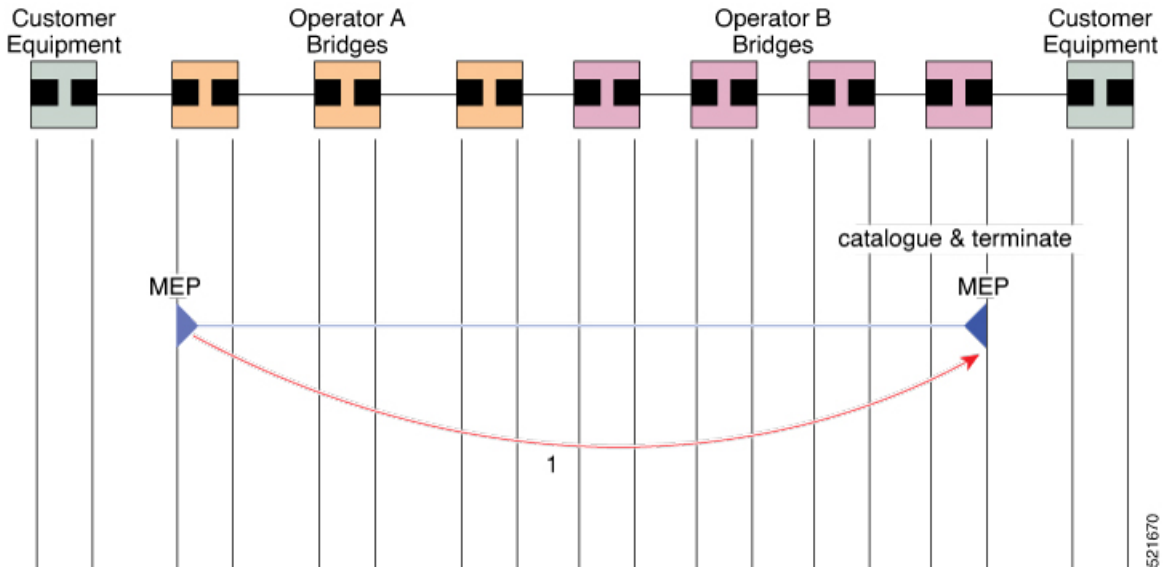
This section describes the following CFM messages:

Continuity Check (IEEE 802.1ag and ITU-T Y.1731)

Continuity Check Messages (CCMs) are "heartbeat" messages exchanged periodically between all the MEPs in a service. Each MEP sends out multicast CCMs, and receives CCMs from all the other MEPs in the service—these are referred to as *peer MEPs*. This allows each MEP to discover its peer MEPs, and to verify that there is connectivity between them.

MIPs also receive CCMs. MIPs use the information to build a MAC learning database that is used when responding to Linktrace. For more information about Linktrace, see the "Linktrace (IEEE 802.1ag and ITU-T Y.1731)" section.

Figure 5: Continuity Check Message Flow



All the MEPs in a service must transmit CCMs at the same interval. IEEE 802.1ag defines the following possible intervals that can be used:

- 100 ms (only supported on bundle members)
- 1 s
- 10 s
- 1 minute
- 10 minutes

A MEP detects a loss of connectivity with one of its peer MEPs when some number of CCMs are missed. This occurs when sufficient time has passed during which a certain number of CCMs were expected, given the CCM interval. This number is called the *loss threshold*, and is usually set to 3.

With the exception of bundle members, CFM is supported only on interfaces that have Layer 2 transport feature enabled.

CCM messages carry a variety of information that allows different defects to be detected in the service. This information includes:

- A configured identifier for the domain of the transmitting MEP. This is referred to as the Maintenance Domain Identifier (MDID).
- A configured identifier for the service of the transmitting MEP. This is referred to as the Short MA Name (SMAN). Together, the MDID and the SMAN make up the Maintenance Association Identifier (MAID). The MAID must be configured identically on every MEP in the service.
- These are restrictions on the type of MAID that are supported for sessions with time interval of less than 1 minute. The MAID supports two types of formats on offloaded MEPs:
    - No Domain Name Format
        - MD Name Format = 1-NoDomainName

- Short MA Name Format = 3 - 2 bytes integer value
- Short MA NAme Length = 2 - fixed length
- Short MA Name = 2 bytes of integer

- 1731 Maid Format
  - MD Name Format = 1-NoDomainName
  - MA Name Format(MEGID Format) = 32
  - MEGID Length = 13 - fixed length
  - MEGID(ICCCode) = 6 Bytes
  - MEGID(UMC) = 7 Bytes
  - ITU Carrier Code (ICC) - Number of different configurable ICC code - 15 (for each NPU)
  - Unique MEG ID Code (UMC) - 4

Maintenance Association Identifier (MAID) comprises of the Maintenance Domain Identifier (MDID) and Short MA Name (SMAN). MDID only supports **null** value and SMAN only supports ITU Carrier Code (ICC) or a numerical. No other values are supported.

- An example for configuring domain ID null is: **ethernet cfm domain SMB level 3 id null**
- An example for configuring SMAN is: **ethernet cfm domain SMB level 3 id null service 901234AB xconnect group 99999 p2p 99999 id number 1**

- A configured numeric identifier for the MEP (the MEP ID). Each MEP in the service must be configured with a different MEP ID.
- Dynamic Remote MEPs are not supported for MEPs with less than 1 min interval. You must configure MEP CrossCheck for all such MEPs.
- Sequence numbering is not supported for MEPs with less than 1 minute interval.
- In a Remote Defect Indication (RDI), each MEP includes this in the CCMs it is sending, if it has detected a defect relating to the CCMs it is receiving. This notifies all the MEPs in the service that a defect has been detected somewhere in the service.
- The interval at which CCMs are being transmitted.
- CCM Tx/Rx statistics counters are not supported for MEPs with less than1 minute intervals.
- Sender TLV and Cisco Proprietary TLVs are not supported for MEPs with less than 1 minute intervals.
- The status of the interface where the MEP is operating, for example, whether the interface is up, down, STP blocked, and so on.

Note The status of the interface (up/down) should not be confused with the direction of any MEPs on the interface (Up MEPs/Down MEPs).

These defects can be detected from the received CCMs:

- Interval mismatch: The CCM interval in the received CCM does not match the interval that the MEP is sending CCMs.
- Level mismatch: A MEP has received a CCM carrying a lower maintenance level than the MEPs own level.
- Loop: A CCM is received with the source MAC address equal to the MAC address of the interface where the MEP is operating.
- Configuration error: A CCM is received with the same MEP ID as the MEP ID configured for the receiving MEP.
- Cross-connect: A CCM is received with a MAID that does not match the locally configured MAID. This generally indicates a VLAN misconfiguration within the network, such that CCMs from one service are leaking into a different service.
- Peer interface down: A CCM is received that indicates the interface on the peer is down.
- Remote defect indication: A CCM is received carrying a remote defect indication.

Note This defect does not cause the MEP to include a remote defect indication in the CCMs that it is sending.

Out-of-sequence CCMs can also be detected by monitoring the sequence number in the received CCMs from each peer MEP. However, this is not considered a CCM defect.

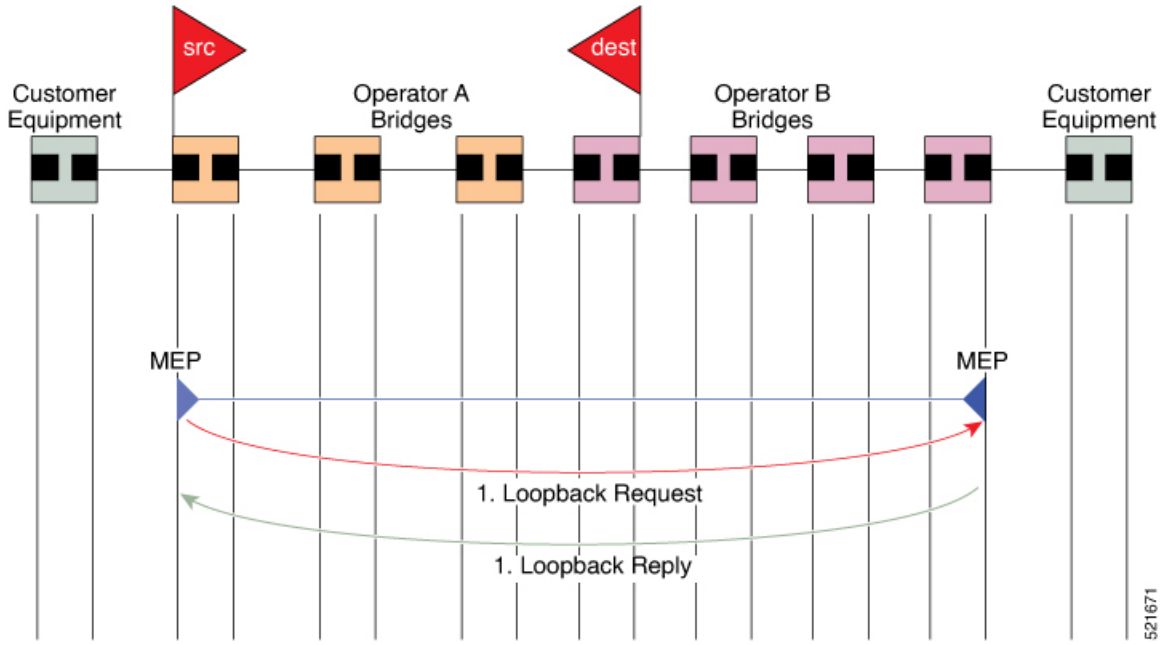
Loopback (IEEE 802.1ag and ITU-T Y.1731)

Loopback Messages (LBM) and Loopback Replies (LBR) are used to verify connectivity between a local MEP and a particular remote MP. At the request of the administrator, a local MEP sends unicast LBMs to the remote MP.

On receiving each LBM, the target maintenance point sends an LBR back to the originating MEP. Loopback indicates whether the destination is reachable or not—it does not allow hop-by-hop discovery of the path. It is similar in concept to an ICMP Echo (ping). Since loopback messages are destined for unicast addresses, they are forwarded like normal data traffic, while observing the maintenance levels. At each device that the loopback reaches, if the outgoing interface is known (in the bridge's forwarding database), then the frame is sent out on that interface. If the outgoing interface is not known, then the message is flooded on all interfaces.

This figure shows an example of CFM loopback message flow between a MEP and MEP.

Figure 6: Loopback Messages



Loopback messages can be padded with user-specified data. This allows data corruption to be detected in the network. They also carry a sequence number which allows for out-of-order frames to be detected.

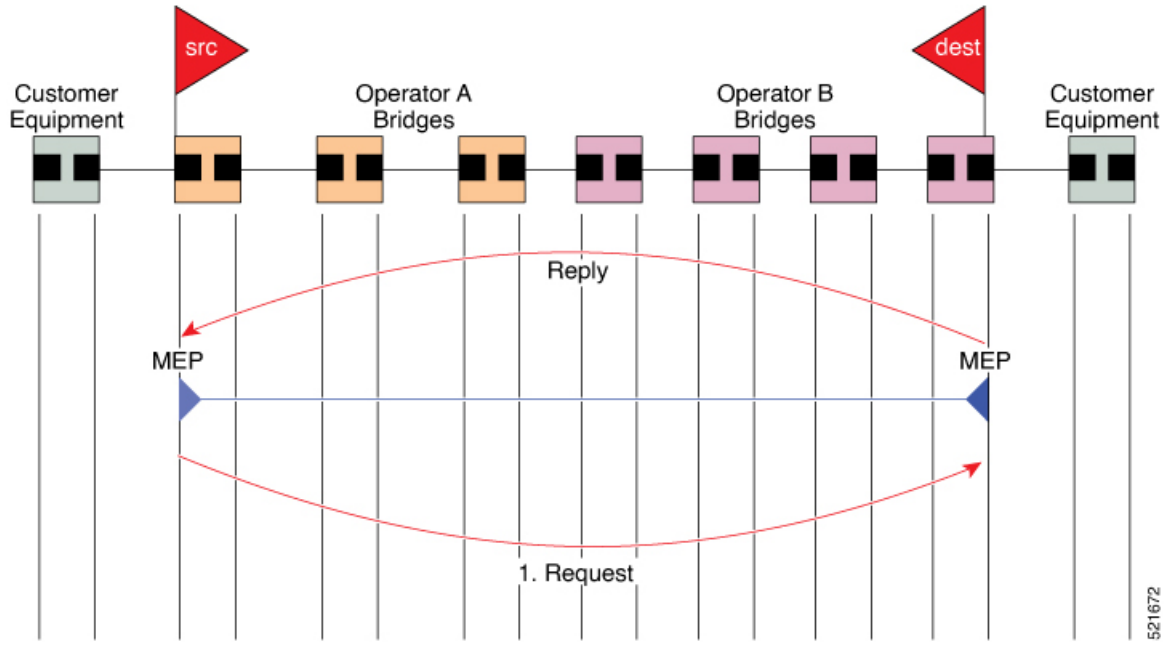
Linktrace (IEEE 802.1ag and ITU-T Y.1731)

Linktrace Messages (LTM) and Linktrace Replies (LTR) are used to track the path (hop-by-hop) to a unicast destination MAC address.

At the request of the operator, a local MEP sends an Linktrace Messages (LTM). Each hop where there is a maintenance point sends an Linktrace Replies (LTR) back to the originating MEP. This allows the administrator to discover connectivity data about the path. It is similar in concept to IP traceroute, although the mechanism is different. In IP traceroute, successive probes are sent, whereas CFM Linktrace uses a single LTM which is forwarded by each MP in the path. LTMs are multicast, and carry the unicast target MAC address as data within the frame. They are intercepted at each hop where there is a maintenance point, and either retransmitted or dropped to discover the unicast path to the target MAC address.

This figure shows an example of CFM linktrace message flow between MEPs and MEPs.

Figure 7: Linktrace Message Flow



The linktrace mechanism is designed to provide useful information even after a network failure. This allows it to be used to locate failures, for example after a loss of continuity is detected. To achieve this, each MP maintains a CCM Learning Database. This maps the source MAC address for each received CCM to the interface through which the CCM was received. It is similar to a typical bridge MAC learning database, except that it is based only on CCMs and it times out much more slowly—on the order of days rather than minutes.

Note In IEEE 802.1ag, the CCM Learning Database is referred to as the MIP CCM Database. However, it applies to both MIPs and MEPs.

In IEEE 802.1ag, when an MP receives an LTM message, it determines whether to send a reply using the following steps:

1. The target MAC address in the LTM is looked up in the bridge MAC learning table. If the MAC address is known, and therefore the egress interface is known, then an LTR is sent.
2. If the MAC address is not found in the bridge MAC learning table, then it is looked up in the CCM learning database. If it is found, then an LTR is sent.
3. If the MAC address is not found, then no LTR is sent (and the LTM is not forwarded).

If the target MAC has never been seen previously in the network, the linktrace operation will not produce any results.

Note IEEE 802.1ag and ITU-T Y.1731 define slightly different linktrace mechanisms. In particular, the use of the CCM learning database and the algorithm described above for responding to LTM messages are specific to IEEE 802.1ag. IEEE 802.1ag also specifies additional information that can be included in LTRs. Regardless of the differences, the two mechanisms are interoperable.

Configurable Logging

CFM supports logging of various conditions to syslog. Logging can be enabled independently for each service, and when the following conditions occur:

- New peer MEPs are detected, or loss of continuity with a peer MEP occurs.
- Changes to the CCM defect conditions are detected.
- Cross-check “missing” or “unexpected” conditions are detected.
- AIS condition detected (AIS messages received) or cleared (AIS messages no longer received).
- EFD used to shut down an interface, or bring it back up.

How to Configure Ethernet OAM

This section provides these configuration procedures:

Configuring Ethernet OAM

Custom EOAM settings can be configured and shared on multiple interfaces by creating an EOAM profile in Ethernet configuration mode and then attaching the profile to individual interfaces. The profile configuration does not take effect until the profile is attached to an interface. After an EOAM profile is attached to an interface, individual EOAM features can be configured separately on the interface to override the profile settings when desired.

This section describes how to configure an EOAM profile and attach it to an interface in these procedures:

Configuring an Ethernet OAM Profile

Perform these steps to configure an Ethernet OAM profile.

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**ethernet oam profile** *profile-name* **Example:** `RP/0/RP0/CPU0:router(config)# ethernet oam profile Profile_1`	Creates a new Ethernet Operations, Administration and Maintenance (OAM) profile and enters Ethernet OAM configuration mode.
Step 3	**link-monitor** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# link-monitor`	Enters the Ethernet OAM link monitor configuration mode.
Step 4	**symbol-period window** *window* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# symbol-period window 60000`	(Optional) Configures the window size (in milliseconds) for an Ethernet OAM symbol-period error event. The IEEE 802.3 standard defines the window size as a number of symbols rather than a time duration. These two formats can be converted either way by using a knowledge of the interface speed and encoding. The range is 1000 to 60000. The default value is 1000.
Step 5	**symbol-period threshold low** *threshold* **high** *threshold* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# symbol-period threshold low 10000000 high 60000000`	(Optional) Configures the thresholds (in symbols) that trigger an Ethernet OAM symbol-period error event. The high threshold is optional and is configurable only in conjunction with the low threshold. The range is 0 to 60000000. The default low threshold is 1.
Step 6		
Step 7	**frame window** *window* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame window 60`	(Optional) Configures the frame window size (in milliseconds) of an OAM frame error event. The range is from 1000 to 60000. The default value is 1000.
Step 8	**frame threshold low** *threshold* **high** *threshold* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame threshold low 10000000 high 60000000`	(Optional) Configures the thresholds (in symbols) that triggers an Ethernet OAM frame error event. The high threshold is optional and is configurable only in conjunction with the low threshold. The range is from 0 to 60000000. The default low threshold is 1.
Step 9	**frame-period window** *window* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period window 60000`	(Optional) Configures the window size (in milliseconds) for an Ethernet OAM frame-period error event. The IEEE 802.3 standard defines the window size as number of frames rather than a time duration. These two formats can be converted either way by using a knowledge of the

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period window milliseconds 60000`	interface speed. Note that the conversion assumes that all frames are of the minimum size. The range is from 100 to 60000. The default value is 1000. **Note** The only accepted values are multiples of the line card-specific polling interval, that is, 1000 milliseconds for most line cards.
Step 10	**frame-period threshold low***threshold* **high** *threshold* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period threshold ppm low 100 high 1000000`	(Optional) Configures the thresholds (in errors per million frames) that trigger an Ethernet OAM frame-period error event. The frame period window is defined in the IEEE specification as a number of received frames, in our implementation it is x milliseconds. The high threshold is optional and is configurable only in conjunction with the low threshold. The range is from 0 to 1000000. The default low threshold is 1. To obtain the number of frames, the configured time interval is converted to a window size in frames using the interface speed. For example, for a 1Gbps interface, the IEEE defines minimum frame size as 512 bits. So, we get a maximum of approximately 1.5 million frames per second. If the window size is configured to be 8 seconds (8000ms) then this would give us a Window of 12 million frames in the specification's definition of Errored Frame Window. The thresholds for frame-period are measured in errors per million frames. Hence, if you configure a window of 8000ms (that is a window of 12 million frames) and a high threshold of 100, then the threshold would be crossed if there are 1200 errored frames in that period (that is, 100 per million for 12 million).
Step 11	**frame-seconds window** *window* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds window 900000`	(Optional) Configures the window size (in milliseconds) for the OAM frame-seconds error event. The range is 10000 to 900000. The default value is 6000. **Note** The only accepted values are multiples of the line card-specific polling interval, that is, 1000 milliseconds for most line cards.

	Command or Action	Purpose
Step 12	**frame-seconds threshold low** *threshold* **high** *threshold* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds threshold low 3 threshold high 900`	(Optional) Configures the thresholds (in seconds) that trigger a frame-seconds error event. The high threshold value can be configured only in conjunction with the low threshold value. The range is 1 to 900 The default value is 1.
Step 13	**exit** **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# exit`	Exits back to Ethernet OAM mode.
Step 14	**mib-retrieval** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# mib-retrieval`	Enables MIB retrieval in an Ethernet OAM profile or on an Ethernet OAM interface.
Step 15	**connection timeout** <*timeout*> **Example:** `RP/0/RP0/CPU0:router(config-eoam)# connection timeout 30`	Configures the connection timeout period for an Ethernet OAM session. as a multiple of the hello interval. The range is 2 to 30. The default value is 5.
Step 16	**hello-interval 1s** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# hello-interval 1s`	Configures the time interval between hello packets for an Ethernet OAM session. The default is 1 second (**1s**).
Step 17	**mode** {**active**|**passive**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# mode passive`	Configures the Ethernet OAM mode. The default is active.
Step 18	**require-remote mode** {**active**|**passive**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# require-remote mode active`	Requires that active mode or passive mode is configured on the remote end before the OAM session becomes active.
Step 19	**require-remote mib-retrieval** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# require-remote mib-retrieval`	Requires that MIB-retrieval is configured on the remote end before the OAM session becomes active.

	Command or Action	Purpose
Step 20	**action capabilities-conflict** {**disable** | **efd** | **error-disable-interface**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action capabilities-conflict efd`	Specifies the action that is taken on an interface when a capabilities-conflict event occurs. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 21	**action critical-event** {**disable** | **error-disable-interface**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action critical-event error-disable-interface`	Specifies the action that is taken on an interface when a critical-event notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 22	**action discovery-timeout** {**disable** | **efd** | **error-disable-interface**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action discovery-timeout efd`	Specifies the action that is taken on an interface when a connection timeout occurs. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 23	**action dying-gasp** {**disable** | **error-disable-interface**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action dying-gasp error-disable-interface`	Specifies the action that is taken on an interface when a dying-gasp notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 24	**action high-threshold** {**error-disable-interface** | **log**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action high-threshold error-disable-interface`	Specifies the action that is taken on an interface when a high threshold is exceeded. The default is to take no action when a high threshold is exceeded. **Note** • If you change the default, the **disable** keyword option is available in Interface Ethernet OAM configuration

	Command or Action	Purpose
		mode to override the profile setting and take no action at the interface when the event occurs.
Step 25	**action session-down** {**disable** | **efd** | **error-disable-interface**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action session-down efd`	Specifies the action that is taken on an interface when an Ethernet OAM session goes down. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 26	**action session-up disable** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action session-up disable`	Specifies that no action is taken on an interface when an Ethernet OAM session is established. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 27	**action uni-directional link-fault** {**disable** | **efd** | **error-disable-interface**}	Specifies the action that is taken on an interface when a link-fault notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 28	**action wiring-conflict** {**disable | efd | log**} **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action session-down efd`	Specifies the action that is taken on an interface when a wiring-conflict event occurs. The default is to put the interface into error-disable state. **Note** • If you change the default, the **error-disable-interface** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and put the interface into error-disable state when the event occurs.
Step 29	**uni-directional link-fault detection** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# uni-directional link-fault detection`	Enables detection of a local, unidirectional link fault and sends notification of that fault to an Ethernet OAM peer.

	Command or Action	Purpose
Step 30	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves the configuration changes to the running configuration file and remains within the configuration session.
Step 31	**end** **Example:** `RP/0/RP0/CPU0:router(config-if)# end`	Ends the configuration session and exits to the EXEC mode.

Attaching an Ethernet OAM Profile to an Interface

Perform these steps to attach an Ethernet OAM profile to an interface:

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure terminal`	Enters global configuration mode.
Step 2	**interface** [**FastEthernet** | **HundredGigE**| **TenGigE**] *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0`	Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port*. **Note** • The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.
Step 3	**ethernet oam** **Example:** `RP/0/RP0/CPU0:router(config-if)# ethernet oam`	Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.
Step 4	**profile** *profile-name* **Example:** `RP/0/RP0/CPU0:router(config-if-eoam)# profile Profile_1`	Attaches the specified Ethernet OAM profile (*profile-name*), and all of its configuration, to the interface.
Step 5	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves the configuration changes to the running configuration file and remains within the configuration session.

	Command or Action	Purpose
Step 6	**end** **Example:** `RP/0/RP0/CPU0:router(config-if)# end`	Ends the configuration session and exits to the EXEC mode.

Configuring Ethernet OAM at an Interface and Overriding the Profile Configuration

Using an EOAM profile is an efficient way of configuring multiple interfaces with a common EOAM configuration. However, if you want to use a profile but also change the behavior of certain functions for a particular interface, then you can override the profile configuration. To override certain profile settings that are applied to an interface, you can configure that command in interface Ethernet OAM configuration mode to change the behavior for that interface.

In some cases, only certain keyword options are available in interface Ethernet OAM configuration due to the default settings for the command. For example, without any configuration of the **action** commands, several forms of the command have a default behavior of creating a syslog entry when a profile is created and applied to an interface. Therefore, the **log** keyword is not available in Ethernet OAM configuration for these commands in the profile because it is the default behavior. However, the **log** keyword is available in Interface Ethernet OAM configuration if the default is changed in the profile configuration so you can retain the action of creating a syslog entry for a particular interface.

To see all of the default Ethernet OAM configuration settings, see the "Verifying the Ethernet OAM Configuration" section.

To configure Ethernet OAM settings at an interface and override the profile configuration, perform these steps:

SUMMARY STEPS

1. **configure**
2. **interface** [HundredGigE | TenGigE] *interface-path-id*
3. **ethernet oam**
4. *interface-Ethernet-OAM-command*
5. **commit**
6. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**interface** [HundredGigE | TenGigE] *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0`	Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port*. **Note** • The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.
Step 3	**ethernet oam** **Example:** `RP/0/RP0/CPU0:router(config-if)# ethernet oam`	Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.
Step 4	*interface-Ethernet-OAM-command* **Example:** `RP/0/RP0/CPU0:router(config-if-eoam)# action capabilities-conflict error-disable-interface`	Configures a setting for an Ethernet OAM configuration command and overrides the setting for the profile configuration, where *interface-Ethernet-OAM-command* is one of the supported commands on the platform in interface Ethernet OAM configuration mode.
Step 5	**commit** **Example:** `RP/0/RP0/CPU0:router(config-if)# commit`	Saves the configuration changes to the running configuration file and remains within the configuration session.
Step 6	**end** **Example:** `RP/0/RP0/CPU0:router(config-if)# end`	Ends the configuration session and exits to the EXEC mode.

Verifying the Ethernet OAM Configuration

Use the **show ethernet oam configuration** command to display the values for the Ethernet OAM configuration for a particular interface, or for all interfaces. The following example shows the default values for Ethernet OAM settings:

```
RP/0/RP0/CPU0:router# show ethernet oam configuration
Thu Aug  5 22:07:06.870 DST
GigabitEthernet0/4/0/0:
  Hello interval:                                        1s
  Mib retrieval enabled:                                  N
  Uni-directional link-fault detection enabled:           N
  Configured mode:                                   Active
  Connection timeout:                                     5
  Symbol period window:                                   0
  Symbol period low threshold:                            1
  Symbol period high threshold:                        None
  Frame window:                                        1000
  Frame low threshold:                                    1
  Frame high threshold:                                None
  Frame period window:                                 1000
  Frame period low threshold:                             1
  Frame period high threshold:                         None
```

```
Frame seconds window:                                 60000
Frame seconds low threshold:                              1
Frame seconds high threshold:                          None
High threshold action:                                 None
Link fault action:                                      Log
Dying gasp action:                                      Log
Critical event action:                                  Log
Discovery timeout action:                               Log
Capabilities conflict action:                           Log
Wiring conflict action:                       Error-Disable
Session up action:                                      Log
Session down action:                                    Log
Require remote mode:                                 Ignore
Require remote MIB retrieval:                             N
```

Configuring Ethernet CFM

To configure Ethernet CFM, perform the following tasks:

Note CFM is not supported for the following:

- L3 Interfaces and Sub-Interfaces
- Bridge Domain, Release 7.3.1 and earlier
- VPLS, Release 7.3.1 and earlier

Configuring a CFM Maintenance Domain

To configure a CFM maintenance domain, perform the following steps:

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
4. **traceroute cache hold-time** *minutes* **size** *entries*
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router(config)# ethernet cfm`	Enters Ethernet Connectivity Fault Management (CFM) configuration mode.
Step 3	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	Creates and names a container for all domain configurations and enters CFM domain configuration mode. The level must be specified. The **id** is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	**traceroute cache hold-time** *minutes* **size** *entries* **Example:** `RP/0/RP0/CPU0:router(config-cfm)# traceroute cache hold-time 1 size 3000`	(Optional) Sets the maximum limit of traceroute cache entries or the maximum time limit to hold the traceroute cache entries. The default is 100 minutes and 100 entries.
Step 5	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring services for a CFM maintenance domain

You can configure up to 50 CFM sessions per line card or 50 CFM sessions per fixed-port router. The system supports 50 CFM sessions on bundles.

Starting Cisco IOS XR Release 7.3.2 and later, 100 CFM sessions are supported for every system.

To configure services for a CFM maintenance domain, perform the following steps:

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
4. **service** *service-name* {**down-meps | xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [[**number** *number*]
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router(config)# ethernet cfm`	Enters Ethernet CFM configuration mode.
Step 3	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	Creates and names a container for all domain configurations at a specified maintenance level, and enters CFM domain configuration mode. The **id** is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	**service** *service-name* {**down-meps	xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*]	[[**number** *number*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service xconnect group X1`	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs. The **id** sets the short MA name.
Step 5	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before`

	Command or Action	Purpose
		`exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling and Configuring Continuity Check for a CFM Service

To configure Continuity Check for a CFM service, complete the following steps:

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
4. **service** *service-name* {**down-meps | xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [[**number** *number*]
5. **continuity-check interval** *time* [**loss-threshold** *threshold*]
6. **continuity-check archive hold-time** *minutes*
7. **continuity-check loss auto-traceroute**
8. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:**	Enters Ethernet Connectivity Fault Management (CFM) configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config)# ethernet cfm`	
Step 3	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The **id** is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	**service** *service-name* {**down-meps** | **xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [[**number** *number*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service xconnect group X1`	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs, or associate the service with a xconnect where up MEPs will be created. The **id** sets the short MA name.
Step 5	**continuity-check interval** *time* [**loss-threshold** *threshold*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check interval 100m loss-threshold 10`	(Optional) Enables Continuity Check and specifies the time interval at which CCMs are transmitted or to set the threshold limit for when a MEP is declared down.
Step 6	**continuity-check archive hold-time** *minutes* **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check archive hold-time 100`	(Optional) Configures how long information about peer MEPs is stored after they have timed out.
Step 7	**continuity-check loss auto-traceroute** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check loss auto-traceroute`	(Optional) Configures automatic triggering of a traceroute when a MEP is declared down.
Step 8	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

	Command or Action	Purpose
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Cross-Check on a MEP for a CFM Service

To configure cross-check on a MEP for a CFM service and specify the expected set of MEPs, complete the following steps:

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
4. **service** *service-name* {**down-meps | xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]]
5. **mep crosscheck**
6. **mep-id** *mep-id-number mep-id-number* [**mac-address** *mac-address*]
7. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router# ethernet cfm`	Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.
Step 3	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]	Creates and names a container for all domain configurations and enters the CFM domain configuration mode.

	Command or Action	Purpose
	Example: `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	The level must be specified. The **id** is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	**service** *service-name* {**down-meps | xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]]	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs, or associate the service with a xconnect where up MEPs will be created. The **id** sets the short MA name.
Step 5	**mep crosscheck** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mep crosscheck mep-id 10`	Enters CFM MEP crosscheck configuration mode.
Step 6	**mep-id** *mep-id-number mep-id-number* [**mac-address** *mac-address*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-xcheck)# mep-id 10`	Enables cross-check on a MEP. **Note** • Repeat this command for every MEP that you want included in the expected set of MEPs for cross-check.
Step 7	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-xcheck)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Other Options for a CFM Service

To configure other options for a CFM service, complete the following steps:

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
4. **service** *service-name* {**down-meps | xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]]
5. **maximum-meps** *number*
6. **log** {**ais**|**continuity-check errors**|**continuity-check mep changes**|**crosscheck errors**|**efd**}
7. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router# ethernet cfm`	Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.
Step 3	**domain** *domain-name* **level** *level-value* [**id** [**null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The **id** is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	**service** *service-name* {**down-meps	xconnect group** *xconnect-group-name* **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*]	[**string** *text*]	[**number** *number*]	[**vlan-id** *id-number*]	[**vpn-id** *oui-vpnid*]]	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs, or associate the service with an xconnect where up MEPs will be created. The **id** sets the short MA name.

	Command or Action	Purpose
Step 5	**maximum-meps** *number* **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# maximum-meps 1000`	(Optional) Configures the maximum number (2 to 8190) of MEPs across the network, which limits the number of peer MEPs recorded in the database.
Step 6	**log** {**ais**|**continuity-check errors**|**continuity-check mep changes**|**crosscheck errors**|**efd**} **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log continuity-check errors`	(Optional) Enables logging of certain types of events.
Step 7	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring CFM MEPs

- For every subinterface configured under a Layer 3 parent interface, you must associate a unique 802.1Q or 802.1ad tag. Else, it leads to unknown network behavior.

SUMMARY STEPS

1. **configure**
2. **interface** {**HundredGigE | TenGigE**} *interface-path-id*
3. **interface** {**HundredGigE | TenGigE | Bundle-Ether**} *interface-path-idl2transport*
4. **ethernet cfm**
5. **mep domain** *domain-name* **service** *service-name* **mep-id** *id-number*

6. **cos** *cos*
7. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface** {**HundredGigE	TenGigE**} *interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1`	Type of Ethernet interface on which you want to create a MEP. Enter **HundredGigE** or **TenGigE** and the physical interface or virtual interface. **Note** • Use the **show interfaces** command to see a list of all interfaces currently configured on the router. • L3 interfaces are only supported for bundle member interfaces. Else, you must enable l2transport.
Step 3	**interface** {**HundredGigE	TenGigE	Bundle-Ether**} *interface-path-idl2transport* **Example:** `RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1`	Type of Ethernet interface on which you want to create a MEP. Enter **HundredGigE, TenGigE,** or **Bundle-Ether** and the physical interface or virtual interface followed by the l2transport. L2transport configures the interface as an L2 interface. Naming convention is *interface-path-id.subinterface*. The period in front of the subinterface value is required as part of the notation.
Step 4	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router(config-if)# ethernet cfm`	Enters interface Ethernet CFM configuration mode.
Step 5	**mep domain** *domain-name* **service** *service-name* **mep-id** *id-number* **Example:** `RP/0/RP0/CPU0:router(config-if-cfm)# mep domain Dm1 service Sv1 mep-id 1`	Creates a maintenance end point (MEP) on an interface and enters interface CFM MEP configuration mode.
Step 6	**cos** *cos* **Example:** `RP/0/RP0/CPU0:router(config-if-cfm-mep)# cos 7`	(Optional) Configures the class of service (CoS) (from 0 to 7) for all CFM packets generated by the MEP on an interface. If not configured, the CoS is inherited from the Ethernet interface.

	Command or Action	Purpose
		Note For Ethernet interfaces, the CoS is carried as a field in the VLAN tag. Therefore, CoS only applies to interfaces where packets are sent with VLAN tags. If the **cos (CFM)** command is executed for a MEP on an interface that does not have a VLAN encapsulation configured, it will be ignored.
Step 7	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-if-cfm-mep)# commit`	Saves configuration changes. • When you use the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Y.1731 AIS

This section has the following step procedures:

Configuring AIS in a CFM Domain Service

Use the following procedure to configure Alarm Indication Signal (AIS) transmission for a CFM domain service and configure AIS logging.

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain** *name* **level** *level*
4. **service** *name* **xconnect group** *xconnect-group-name* **p2p** *xconnect-name*
5. **ais transmission** [**interval** {**1s**|**1m**}][**cos** *cos*]
6. **log ais**
7. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router(config)# ethernet cfm`	Enters Ethernet CFM global configuration mode.
Step 3	**domain** *name* **level** *level* **Example:** `RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1`	Specifies the domain and domain level.
Step 4	**service** *name* **xconnect group** *xconnect-group-name* **p2p** *xconnect-name* **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 xconnect group XG1 p2p X2`	Specifies the service and cross-connect group and name.
Step 5	**ais transmission** [**interval** {**1s**|**1m**}][**cos** *cos*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7`	Configures Alarm Indication Signal (AIS) transmission for a Connectivity Fault Management (CFM) domain service.
Step 6	**log ais** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log ais`	Configures AIS logging for a Connectivity Fault Management (CFM) domain service to indicate when AIS or LCK packets are received.
Step 7	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

	Command or Action	Purpose
		• Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring AIS on a CFM Interface

To configure AIS on a CFM interface, perform the following steps:

SUMMARY STEPS

1. **configure**
2. **interface gigabitethernet** *interface-path-id*
3. **ethernet cfm**
4. **ais transmission up interval 1m cos** *cos*
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface gigabitethernet** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# interface TenGigE 0/0/0/2`	Enters interface configuration mode.
Step 3	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router(config)# ethernet cfm`	Enters Ethernet CFM interface configuration mode.
Step 4	**ais transmission up interval 1m cos** *cos* **Example:**	Configures Alarm Indication Signal (AIS) transmission on a Connectivity Fault Management (CFM) interface.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if-cfm)# ais transmission up interval 1m cos 7`	
Step 5	**end** or **commit** **Example:** `RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit`	Saves configuration changes. • When you issue the **end** command, the system prompts you to commit changes: `Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:` • Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. • Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes. • Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes. • Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

Verifying the CFM Configuration

To verify the CFM configuration, use one or more of the following commands:

show ethernet cfm configuration-errors [**domain** *domain-name*] [**interface** *interface-path-id*]	Displays information about errors that are preventing configured CFM operations from becoming active, as well as any warnings that have occurred.
show ethernet cfm local maintenance-points domain *name* [**service** *name]* | **interface** *type interface-path-id*] [**mep** | **mip**]	Displays a list of local maintenance points.

Note After you configure CFM, the error message, *cfmd[317]: %L2-CFM-5-CCM_ERROR_CCMS_MISSED : Some received CCMs have not been counted by the CCM error counters*, may display. This error message does not have any functional impact and does not require any action from you.

CFM Over Bundles

CFM over bundle supports the following:

- CFM Maintenance Points — UP MEP, Down MEP, which only includes L2 bundle main and sub-interfaces.
- CCM interval of 100 ms, 1 sec, 10 sec, 1 min, and 10 mins.
- RP OIR/VM reload without impacting learnt CFM peer MEPs.
- Process restart without impacting CFM sessions.
- Static MEPs.

Restrictions for Configuration of CFM on Bundles

Following are the restrictions for configuring CFM over bundle member interfaces:

- Only Layer 2 bundle Ethernet interfaces and sub-interfaces are supported, which are part of a L2VPN cross-connect.
- No support for 3.3 ms and 10 ms CCM interval.
- Supports 5000 pps rates of CCM traffic for bundle interfaces.
- Ethernet Connectivity Fault Management (CFM) is not supported with Maintenance association End Points (MEPs) that are configured on default and untagged encapsulated sub-interfaces that are part of a single physical interface.
- Multiple MEPs of different directions are not supported on the same interface or Xconnect.
- CFM does not support fast failover, which may result in session flaps on bundle interfaces. Use offload for virtual interfaces to avoid flaps on faster CCM intervals.

Ethernet SLA Statistics Measurement in a Profile

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
Enhancement to Ethernet SLA Statistics Measurement	Release 7.7.1	You can now configure the size of bins for the delay and jitter measurement in Ethernet SLA statistics with a **width** value ranging from 1 to 10000000 microseconds. This enhancement provides granularity to store more accurate results of SLA statistics in the aggregate bins. In earlier releases, you could only configure the **width** value for the delay and jitter measurement in milliseconds. This feature introduces the **usec** keyword in the **aggregate** command.

The Ethernet SLA feature supports measurement of one-way and two-way delay and jitter statistics, and one-way FLR statistics.

Ethernet SLA statistics measurement for network performance is performed by sending packets and storing data metrics such as:

- Round-trip delay time—The time for a packet to travel from source to destination and back to source again.
- Round-trip jitter—The variance in round-trip delay time (latency).
- One-way delay and jitter—The router also supports measurement of one-way delay or jitter from source to destination, or from destination to source.
- One-way frame loss—The router also supports measurement of one-way frame loss from source to destination, or from destination to source.

In addition to these metrics, these statistics are also kept for SLA probe packets:

- Packet loss count
- Packet corruption event
- Out-of-order event
- Frame Loss Ratio (FLR)

Counters for packet loss, corruption, and, out-of-order packets are kept for each bucket, and in each case, a percentage of the total number of samples for that bucket is reported (for example, 4% packet corruption).

For delay, jitter, and loss statistics, the minimum, maximum, mean and standard deviation for the whole bucket are reported, as well as the individual samples or aggregated bins. Also, the overall FLR for the bucket, and individual FLR measurements or aggregated bins are reported for synthetic loss measurement statistics. The packet loss count is the overall number of measurement packets lost in either direction and the one-way FLR measures the loss in each direction separately.

When aggregation is enabled using the **aggregate** command, bins are created to store a count of the samples that fall within a certain value range, which is set by the **width** keyword. Only a counter of the number of results that fall within the range for each bin is stored. This uses less memory than storing individual results. When aggregation is not used, each sample is stored separately, which can provide a more accurate statistics analysis for the operation, but it is highly memory-intensive due to the independent storage of each sample.

A bucket represents a time period during which statistics are collected. All the results received during that time period are recorded in the corresponding bucket. If aggregation is enabled, each bucket has its own set of bins and counters, and only results relating to the measurements initiated during the time period represented by the bucket are included in those counters.

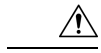
Frame Loss Ratio (FLR) is a primary attribute that can be calculated based on loss measurements. FLR is defined by the ratio of lost packets to sent packets and expressed as a percentage value. FLR is measured in each direction (source to destination and destination to source) separately. Availability is an attribute that is typically measured over a long period of time, such as weeks or months. The intent is to measure the proportion of time when there was prolonged high loss.

To configure one-way delay or jitter measurements, you must first configure the **profile (SLA)** command using the **type cfm-delay-measurement** form of the command.

For valid one-way delay results, you must have both local and remote devices time synchronized. In order to do this, you must select sources for frequency and time-of-day (ToD).

Frequency selection can be between any source of frequency available to the router, such as: BITS, GPS, SyncE, or PTP. The ToD selection is between the source selected for frequency and PTP or DTI. Note that NTP is not sufficient.

Configuration Guidelines



Caution Certain SLA configurations can use a large amount of memory which can affect the performance of other features on the router.

Before you configure Ethernet SLA, consider the following guidelines:

- Aggregation—Use of the **aggregate none** command significantly increases the amount of memory required because each individual measurement is recorded, rather than just counts for each aggregation bin. When you configure aggregation, consider that more bins will require more memory.
- Buckets archive—When you configure the **buckets archive** command, consider that the more history that is kept, the more memory will be used.
- Measuring two statistics (such as both delay and jitter) will use approximately twice as much memory as measuring one.
- Separate statistics are stored for one-way source-to-destination and destination-to-source measurements, which consumes twice as much memory as storing a single set of round-trip statistics.

- You must define the schedule before you configure SLA probe parameters to send probes for a particular profile. It is recommended to set up the profile—probe, statistics, and schedule before any commit.

Restrictions

One-way delay and jitter measurements are not supported by cfm-loopback profile types.

Configure Ethernet SLA Statistics Measurement in a Profile

To configure SLA statistics measurement in a profile, perform these steps:

1. Enter the Ethernet SLA configuration mode, using the **ethernet sla** command in Global Configuration mode.
2. Create an SLA operation profile with the **profile** *profile-name* **type cfm-delay-measurement** command.
3. Enable the collection of SLA statistics using the **statistics measure** {**one-way-delay-ds | one-way-delay-sd | one-way-jitter-ds | one-way-jitter-sd | round-trip-delay | round-trip-jitter | one-way-loss-ds | one-way-loss-sd**} command.
4. Configure the size and number of bins into which to aggregate the results of statistics collection. For delay measurements and data loss measurements, the default is that all values are aggregated into 1 bin. For synthetic loss measurements, by default the aggregation is disabled. Use the **aggregate** {**bins** *count* **width** [**usec**] *width* | **none**} command to configure the bins.
   - For delay and jitter measurements, you can configure a width value from 1 to 10000 milliseconds, if the number of bins is at least 2. To configure the width value in microseconds, use the **usec** option. You can configure the width value from 1 to 10000000 microseconds.
   - For data loss and synthetic loss measurements, you can configure a width value from 1 to 100 percentage points, if the number of bins is at least 2.
5. Configure the size of the buckets in which statistics are collected, using the **buckets size** *number* **probes** command.
6. Configure the number of buckets to store in memory using the **buckets archive** *number* command.
7. Save the configuration changes using the **end** or **commit** command.

Configuration Example

This example shows configuration of round-trip-delay statistics measurement in 5 bins each with a range of 123 microseconds:

```
Router(config)# ethernet sla
Router(config-sla)# profile test type cfm-delay-measurement
Router(config-sla-prof)# statistics measure round-trip-delay
Router(config-sla-prof-stat-cfg)# aggregate bins 5 width usec 123
Router(config-sla-prof-stat-cfg)# buckets size 1 probes
Router(config-sla-prof-stat-cfg)# buckets archive 50
Router(config-sla-prof-stat-cfg)# commit
```

This example shows configuration of round-trip-delay statistics measurement in 5 bins each with a range of 10 milliseconds:

```
Router(config)# ethernet sla
Router(config-sla)# profile test type cfm-delay-measurement
Router(config-sla-prof)# statistics measure round-trip-delay
Router(config-sla-prof-stat-cfg)# aggregate bins 5 width 10
Router(config-sla-prof-stat-cfg)# buckets size 1 probes
Router(config-sla-prof-stat-cfg)# buckets archive 50
Router(config-sla-prof-stat-cfg)# commit
```

Verification

This example displays aggregate bins configured with a range of 123 microseconds:

```
Router# show ethernet sla statistics detail
Tue Sep 28 07:59:22.340 PDT
Source: Interface GigabitEthernet0/0/0/2, Domain dom1
Destination: Target MAC Address 0012.0034.0056
================================================================================
Profile 'test', packet type 'cfm-delay-measurement'
Scheduled to run every 1min first at 00:00:31 UTC for 10s

Round Trip Delay
~~~~~~~~~~~~~~~~
1 probes per bucket

No stateful thresholds.

Bucket started at 07:56:31 PDT Tue 28 September 2021 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                  Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 10
    Min: 0.000ms, occurred at 07:56:32 PDT Tue 28 September 2021
    Max: 1.000ms, occurred at 07:56:31 PDT Tue 28 September 2021
    Mean: 0.100ms; StdDev: 0.300ms

    Bins:
    Range                 Samples   Cum. Count     Mean
    -----------------  ----------  -----------  -------
        0 to 0.123 ms  9  (90.0%)   9  (90.0%)  0.000ms
    0.123 to 0.246 ms  0   (0.0%)   9  (90.0%)        -
    0.246 to 0.369 ms  0   (0.0%)   9  (90.0%)        -
    0.369 to 0.492 ms  0   (0.0%)   9  (90.0%)        -
    > 0.492        ms  1  (10.0%)  10 (100.0%)  1.000ms
```

This example displays aggregate bins configured with a range of 10 milliseconds:

```
Router# show ethernet sla statistics detail
Tue Sep 28 08:00:57.527 PDT
Source: Interface GigabitEthernet0/0/0/2, Domain dom1
Destination: Target MAC Address 0012.0034.0056
================================================================================
Profile 'test', packet type 'cfm-delay-measurement'
Scheduled to run every 1min first at 00:00:31 UTC for 10s

Round Trip Delay
~~~~~~~~~~~~~~~~
1 probes per bucket

No stateful thresholds.

Bucket started at 08:00:32 PDT Tue 28 September 2021 lasting 10s
    Pkts sent: 9; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                  Misordered: 1 (11.1%); Duplicates: 0 (0.0%)
    Result count: 9
```

```
Min: 0.000ms, occurred at 08:00:32 PDT Tue 28 September 2021
Max: 0.000ms, occurred at 08:00:32 PDT Tue 28 September 2021
Mean: 0.000ms; StdDev: 0.000ms

Results suspect due to a probe starting mid-way through a bucket

Bins:
Range           Samples  Cum. Count     Mean
-----------  ----------  ----------  -------
 0 to 10 ms  9 (100.0%)  9 (100.0%)  0.000ms
10 to 20 ms  0   (0.0%)  9 (100.0%)        -
20 to 30 ms  0   (0.0%)  9 (100.0%)        -
30 to 40 ms  0   (0.0%)  9 (100.0%)        -
> 40     ms  0   (0.0%)  9 (100.0%)        -
```

Ethernet frame delay measurement for L2VPN services

Ethernet frame delay measurement complies with the ITU-T Y.1731 standard, which provides comprehensive fault management and performance monitoring recommendations. Delay Measurement Message (DMM) and Delay Measurement Reply (DMR) are used to periodically measure one-way or two-way frame delay and frame delay variation between a pair of point-to-point MEPs. Measurements are made between two MEPs belonging to the same domain and Maintenance Association (MA).

Table 5: Feature History Table

Feature Name	Release Information	Feature Description
Ethernet frame delay measurement for L2VPN services	Release 7.5.3	You can now monitor L2VPN networks and avoid impact to your customers' operations by accurately measuring frame round-trip delays and jitters between two maintenance endpoints (MEPs). This feature lets you detect end-to-end connectivity, loopback, and link trace on MEPs. It reports service performance to your end customers, helping improve technical and operational tasks such as troubleshooting and billing. This feature introduces the **cfm-delay-measurement probe** command.

You can measure frame delay in the Layer 2 networks to detect end-to-end connectivity, loopback, and link trace on Maintenance End Points (MEPs) and also report service performance that helps to improve technical and operational tasks such as troubleshooting, billing, and so on. Frame delay is the duration between the time the source node transmits the first bit of a frame and the time the same source node receives the last bit of the frame.

The frame delay measurement uses the following two protocol data units (PDUs):

- Delay Measurement Message (DMM)—DMM is used to measure frame delay and frame delay variation between a pair of point-to-point Maintenance End Points (MEPs).
- Delay Measurement Response (DMR)—DMR is the delay measurement response sent by the destination MEP. When an MEP receives a DMM frame, the responder MEP responds with a DMR frame. The DMR frame carries a reply information and a copy of the timestamp contained in the DMM frame.

Note DMM sessions (using CFM) are not supported with MACsec enabled on the core interface, as this requires pre-encryption timestamping in the interface group.

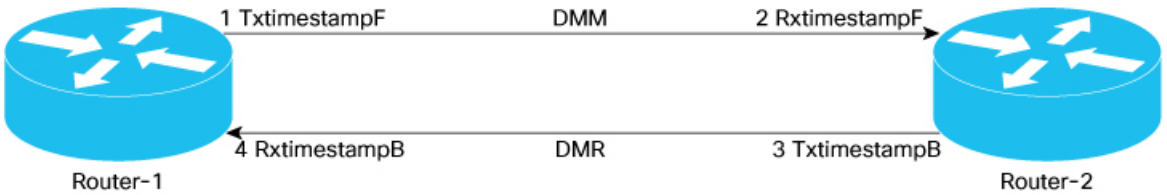
We support one-way and two-way frame delay measurement.

Frame Delay Measurement	Description
One-way frame delay measurement (1DM)	• Measures the frame delay on a unidirectional link between the MEPs. • 1DM requires that clocks at both the transmitting MEP and the receiving MEPs are synchronized. • Measuring frame-delay variation does not require clock synchronization and the variation can be measured using 1DM and DMR frame combination.
Two-way frame delay measurement	• Measures the frame delay on a bidirectional link between the MEPs. • Two-way delay measurement does not require the clocks at both the transmitting MEP and the receiving MEPs to be synchronized. • The two-way frame delay is measured using only DMM and DMR frames.

For more information about CFM, see Configuring Ethernet OAM, on page 1.

Topology

Let's see how a round-trip frame delay is measured with the following sample topology.



- The sender MEP (Router-1) transmits a frame containing delay measurement request information and the timestamp at the which router sends the DMM.

- When packets pass through each interface, timestamps are written into DMMs and DMRs at both local and peer MEPs.
- When the DMM leaves the local interface, the TX timestamp is added to the packet.
- When the receiver MEP (Router-2) receives the frame, records the timestamp at which the receiver MEP receives the frame with the delay measurement request information and the remote MEP (Router-2) responds with an DMR adding the remote TX timestamp to the packet as it leaves the remote interface.

To measure a round-trip delay for a traffic exchange between Router-1 and Router-2, four timestamps get populated as the packet moves through the network.

- Router-1 adds the TxTimestampF when DMM packet is transmitted.
- Router-2 adds RxTimestampF when DMM packet is received by it.
- Router-2 adds TxTimestampB when DMR packet it transmitted.
- Router-1 adds RxTimestampB when DMR is received by it

The round-trip delay is calculated using the following formula:

```
Delay = (RxTimestampB – TxTimestampF) - (TxTimestampB – RxTimestampF)
    = RxTimestampB – TxTimestampF – TxTimestampB + RxTimestampF
    = (RxTimestampF – TxTimestampF) - (TxTimestampB – RxTimestampB)
```

Configure Ethernet Frame Delay Measurement for L2VPN Services

Perform the following tasks to configure Ethernet Frame Delay Measurement for L2VPN Services:

1. Configure L2VPN service.
2. Enable CFM service continuity check.
3. Enable CFM on the interface.
4. Configure Ethernet frame delay measurement.

```
/* Configure L2VPN service */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group evpn_vpws_203
Router(config-l2vpn-xc)# p2p evpn_vpws_phy-100
Router(config-l2vpn-xc-p2p)# interface GigabitEthernet0/0/0/2.100
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 30001 target 30001 source 50001
Router(config-l2vpn-xc-p2p)# commit

/* Enable CFM service continuity check */
Router# ethernet cfm
Router(config-cfm# domain xcup1 level 7 id null
Router(config-cfm-dmn)# service xcup1 xconnect group evpn_vpws_Bund
Router(config-cfm-dmn-svc)# mip auto-create all ccm-learning
Router(config-cfm-dmn-svc)# continuity-check interval 1s
Router(config-cfm-dmn-svc)# mep crosscheck
Router(config-cfm-dmn-svc)# mep-id 4001
Router(config-cfm-dmn-svc)# commit

/* Enable CFM on the interface */
Router(config)# interface GigabitEthernet0/0/0/2.100 l2transport
```

```
Router(config-subif)# encapsulation dot1q 100
Router(config-subif)# rewrite ingress tag pop 1 symmetric
Router(config-subif)# mtu 9100
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain bd-domain service bd-service mep-id 4001
Router(config-if-cfm-mep)# sla operation profile test-profile1 target mep-id 1112
Router(config-if-cfm-mep)# commit

/* Configure Ethernet frame delay measurement */
Router(config)# ethernet sla
Router(config-sla)# profile EVC-1 type cfm-delay-measurement
Router(config-sla-prof)# probe
Router(config-sla-prof-pb)# send packet every 1 seconds
Router(config-sla-prof-pb)# schedule
Router(config-sla-prof-schedule)# every 3 minutes for 120 seconds
Router(config-sla-prof-schedule)# statistics
Router(config-sla-prof-stat)# measure round-trip-delay
Router(config-sla-prof-stat-cfg)# buckets size 1 probes
Router(config-sla-prof-stat-cfg)# buckets archive 5
Router(config-sla-prof-stat-cfg)# commit
```

Running Configuration

This section shows the Ethernet frame delay measurement running configuration.

```
/* Configure L2VPN service */
l2vpn
xconnect group evpn_vpws_203
p2p evpn_vpws_phy-100
interface GigabitEthernet0/0/0/2.100
neighbor evpn evi 30001 target 30001 source 50001
!
/* Enable CFM service continuity check */
ethernet cfm
 domain xcup1 level 7 id null
  service xcup1 xconnect group evpn_vpws_Bundle_ether203 p2p evpn_vpws-100 id number 4001
   mip auto-create all ccm-learning
   continuity-check interval 1s
   mep crosscheck
    mep-id 4001
!
/* Enable CFM on the interface */
interface GigabitEthernet0/0/0/2.100 l2transport
 encapsulation dot1q 100
 rewrite ingress tag pop 1 symmetric
 mtu 9100
 ethernet cfm
  mep domain bd-domain service bd-service mep-id 4001
   sla operation profile test-profile1 target mep-id 1112
!
/* Configure Ethernet SLA */
ethernet sla
 profile EVC-1 type cfm-delay-measurement
  probe
   send packet every 1 seconds
  !
  schedule
   every 3 minutes for 120 seconds
  !
  statistics
   measure round-trip-delay
    buckets size 1 probes
```

```
    buckets archive 5
!
```

Verification

Verify the frame delay measurement. In the following example, you observe that the sent and received DMM and DMR packets are same. So there is no delay in frame transimission.

```
Router# show ethernet cfm local meps interface GigabitEthernet0/0/0/2.100 verbose

Up MEP on GigabitEthernet0/0/0/2.100 MEP-ID 4001
==============================================================================
  Interface state: Up      MAC address: 0c11.6752.3af8
  Peer MEPs: 1 up, 0 with errors, 0 timed out (archived)

  CCM generation enabled:  Yes, 10s (Remote Defect detected: No)
  AIS generation enabled:  No
  Sending AIS:             No
  Receiving AIS:           No
  Sending CSF:             No
  Receiving CSF:           No

  Packet        Sent      Received
  ------  ----------  ------------------------------------------------------------
   CCM          19      9 (out of seq: 0)
DMM          473     0
DMR          0       473
```