

Configure Ethernet OAM

This chapter describes the Cisco IOS XR commands to configure Ethernet OAM.

Table 1: Feature History

Feature Name	Release Information	Feature Description
Delay Measurement	Cisco IOS XR Release 6.5.31	This feature supports hardware timestamping for both one-way and two-way delay measurements. The delay measurement can be performed only after the Precision Time Protocol (PTP) is configured and the clocks are in sync. The delay measurement packets contain timestamps within the packet data that can be used to accurately measureframe delay, jitter, and round-trip statistics.

- Understanding PTP, on page 1
- Prerequisites for Configuring Ethernet OAM, on page 5
- Restrictions for Configuring Ethernet OAM, on page 5
- Information About Configuring Ethernet OAM, on page 6
- How to Configure Ethernet OAM, on page 31
- Configuration Examples for Ethernet OAM, on page 61

Understanding PTP

Precision Time Protocol (PTP) is a protocol that defines a method to distribute time around a network. PTP support is based on the IEEE 1588-2008 standard. PTP synchronizes with the real-time clocks of the devices in a network at nanosecond accuracy. The clocks are organized into a primary-secondary hierarchy. PTP identifies the port that is connected to a device with the most precise clock. This clock is referred to as the primary clock. All the other devices on the network synchronize their clocks with the primary clock and are referred to as members. Constantly exchanged timing messages ensure continued synchronization.

Note When PTP switching happens between interfaces, internal servo status might move to Freerun and restore to PHASE_LOCKED state. This is expected in NCS 4000.

ITU-T Telecom Profiles for PTP

Cisco IOS XR software supports ITU-T Telecom Profiles for PTP as defined in the ITU-T recommendations. A profile is a specific selection of PTP configuration options that are selected to meet the requirements of a particular application.

PTP lets you define separate profiles to adapt itself for use in different scenarios. A telecom profile differs in several ways from the default behavior defined in the IEEE 1588-2008 standard.

G.8275.1 ITU-T telecom profile is supported for PTP.

G.8275.1

G.8275.1 profile fulfills the time-of-day and phase synchronization requirements in telecom networks with all network devices participating in the PTP protocol. G.8275.1 profile provides better frequency stability for the time-of-day and phase synchronization.

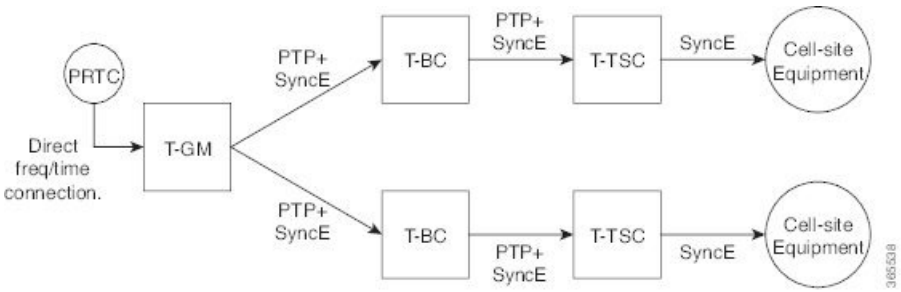
Features of G.8275.1 profile are:

- Synchronization Model: G.8275.1 profile adopts hop-by-hop synchronization model. Each network device in the path from primary to secondary synchronizes its local clock to upstream devices and provides synchronization to downstream devices.
- Clock Selection: G.8275.1 profile also defines an alternate BMCA that selects a clock for synchronization and port state for the local ports of all devices in the network is defined for the profile. The parameters defined as a part of the BMCA are:
    - Clock Class
    - Clock Accuracy
    - Offset Scaled Log Variance
    - Priority 2
    - Clock Identity
    - Steps Removed
    - Port Identity
    - notSlave flag
    - Local Priority
- Port State Decision: The port states are selected based on the alternate BMCA algorithm. A port is configured to a primary-only port state to enforce the port to be a primary for multicast transport mode.
- Packet Rates: The nominal packet rate for Announce packets is 8 packets-per-second and 16 packets-per-second for Sync/Follow-Up and Delay-Request/Delay-Response packets.
- Transport Mechanism: G.8275.1 profile supports only Ethernet PTP transport mechanism.

- Mode: G.8275.1 profile supports transport of data packets only in multicast mode. The forwarding is done based on forwardable or non-forwardable multicast MAC address.
- Domain Numbers: The domain numbers that can be used in a G.8275.1 profile network ranges from 24 to 43. The default domain number is 24.
- Clock Type: G.8275.1 profile supports the following clock types:
  - T-GM: The telecom grandmaster (T-GM) provides timing to all other devices on the network. It does not synchronize its local clock with any other network element other than the Primary Reference Time Clock (PRTC).
  - T-BC: The telecom boundary clock (T-BC) synchronizes its local clock to a T-GM or an upstream T-BC and provides timing information to downstream T-BCs or T-TSCs. If at a given point in time there are no higher-quality clocks available to a T-BC to synchronize to, it may act as a grandmaster.
  - T-TSC: The telecom time slave clock (T-TSC) synchronizes its local clock to another PTP clock (in most cases, the T-BC), and does not provide synchronization through PTP to any other device.

The following figure describes a sample G.8275.1 topology.

Figure 1: A Sample G.8275.1 Topology



Configuring Global G.8275.1 Profile

The following configuration describes the steps involved to create a global configuration profile for a PTP interface that can then be assigned to any interface as required. It uses G.8275.1 profile as an example

Procedure

Example:

```
RP/0/RP0/CPU0:router# config terminal
RP/0/RP0/CPU0:router(config)# ptp
RP/0/RP0/CPU0:router(config-ptp)# clock
RP/0/RP0/CPU0:router(config-ptp-clock)# domain 24
RP/0/RP0/CPU0:router(config-ptp-clock)# profile g.8275.1 clock-type T-BC
RP/0/RP0/CPU0:router(config-ptp-clock)# exit
RP/0/RP0/CPU0:router(config-ptp)# profile slave
RP/0/RP0/CPU0:router(config-ptp-profile)# multicast target-address ethernet 01-1B-19-00-00-00
RP/0/RP0/CPU0:router(config-ptp-profile)# transport ethernet
RP/0/RP0/CPU0:router(config-ptp-profile)# sync frequency 16
RP/0/RP0/CPU0:router(config-ptp-profile)# announce frequency 8
```

```
RP/0/RP0/CPU0:router(config-ptp-profile)# delay-request frequency 16
RP/0/RP0/CPU0:router(config-ptp-profile)# exit
RP/0/RP0/CPU0:router(config-ptp)# profile master
RP/0/RP0/CPU0:router(config-ptp-profile)# multicast target-address ethernet 01-1B-19-00-00-00
RP/0/RP0/CPU0:router(config-ptp-profile)# transport ethernet
RP/0/RP0/CPU0:router(config-ptp-profile)# sync frequency 16
RP/0/RP0/CPU0:router(config-ptp-profile)# announce frequency 8
RP/0/RP0/CPU0:router(config-ptp-profile)# delay-request frequency 16
RP/0/RP0/CPU0:router(config-ptp-profile)# exit
RP/0/RP0/CPU0:router(config-ptp)# physical-layer-frequency
RP/0/RP0/CPU0:router(config-ptp)# log
RP/0/RP0/CPU0:router(config-ptp-log)# servo events
RP/0/RP0/CPU0:router(config-ptp-log)# commit
```

Configuring PTP Telecom Profile Clock

This procedure describes the steps involved to configure PTP clock and its settings to be consistent with ITU-T Telecom Profiles for Frequency.

Procedure

Step 1 **ptp**

Example:

```
RP/0/RP0/CPU0:router(config)# ptp
```

Enters the PTP configuration mode.

Step 2 **clock**

Example:

```
RP/0/RP0/CPU0:router(config-ptp)# clock
```

Enters the PTP-clock configuration mode.

Step 3 **domain** *domain-number*

Example:

```
RP/0/RP0/CPU0:router(config-ptp)# domain 24
```

Configures the domain number for a PTP profile. The allowed domain number range for G.8275.1 profile is between 24 and 43.

Step 4 **exit**

Example:

```
RP/0/RP0/CPU0:router(config-ptp-clock)# exit
```

Exits the PTP-clock configuration mode.

Step 5 **clock profile g.8275.1 clock-type** *clock-type*

Example:

```
RP/0/RP0/CPU0:router(config-ptp-clock)# clock profile g.8275.1 clock-type T-GM
```

Configures the desired telecom profile and the clock type for the profile.

Prerequisites for Configuring Ethernet OAM

Before configuring Ethernet OAM, confirm that at least one of the Ethernet line cards is installed on the router.

- NCS4K-2H10T-OP-KS
- NCS4K-4H-OPW-QC2

Restrictions for Configuring Ethernet OAM

Table 2: Feature History

Feature Name	Release Information	Feature Description
Hardware Timestamping Support on Multi-chassis	Cisco IOS XR Release 6.5.32	Hardware timestamping is supported for both one-way and two-way delay measurements in multi-chassis.

The following functional areas of Ethernet OAM are not supported:

- CFM is not supported on dot1q second-dot1q any and dot1ad dot1q any.
- CFM is not supported for offload session which does not have short MA name.
- Sender-ID TLV is not supported for offloaded session.
- From R6.5.28, CFM supports Y.1731 Performance Measurement i.e Delay Measurement(DMM), Loss Measurement(LMM) and Synthetic Measurement(SLM).
- From 6.5.28, software timestamping is supported for both one-way and two-way delay measurements.
- From 6.5.31, hardware timestamping is supported for both one-way and two-way delay measurements on a single chassis.
- From 6.5.32, hardware timestamping is supported for both one-way and two-way delay measurements on multi-chassis.
- CFM down-meps are not supported on L3 interface.
- CFM does not support rewrite scenarios for vlan defaults and for translate 2 to 1.
- CFM does not support MIP CCM Learning.
- CFM on Cisco IOS XR Software does not support a tag stack of more than two tags.
- If a subinterface is configured that matches untagged Ethernet frames (for example, by configuring the **encapsulation default** command), then you can not create a down MEP on the underlying physical or bundle interface.

- Both up MEPs and down MEPs are not supported on Layer 3 interfaces.
- While performing RPVM Switch Over or RP OIR or ISSU, the packet transmission stops for a duration of 3 to 20 seconds and causes EOAM session to flap (session goes down and recovers back).
- From Release 6.5.35, these restrictions apply:
  - CFM supports up to two VLAN tags per packet. This matches the limitation on the DNX platform, which also supports a maximum of two VLAN tags. Packets with three or more VLAN tags are not supported.
  - CFM does not support dot1q or dot1ad encapsulation when using the "any" option.
  - CFM cannot be configured on Layer 2 sub-interfaces derived from the same main interface if one uses the default encapsulation and the other uses untagged encapsulation. Since both default and untagged configurations lack VLAN tags, packet injection is restricted, and CFM is not supported on these interfaces.

Information About Configuring Ethernet OAM

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

Ethernet Link OAM

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, . 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 EOAM 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.

Ethernet CFM

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.

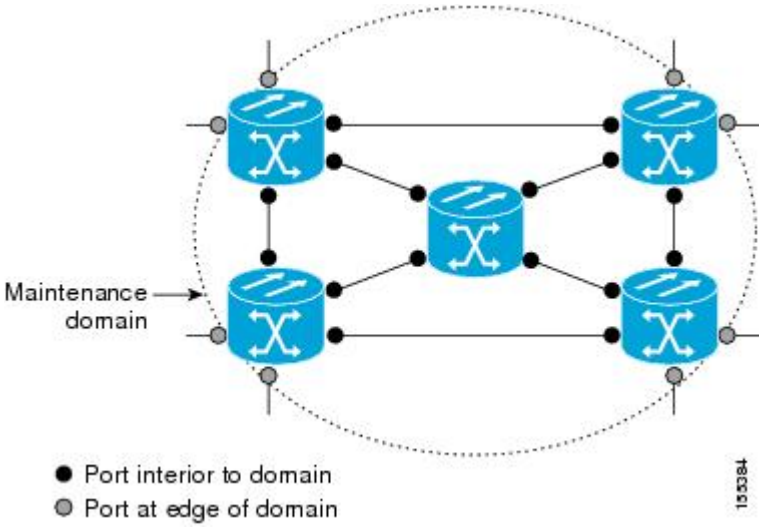
- ETH-AIS—The reception of ETH-LCK messages is also supported.

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

Maintenance Domains

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:

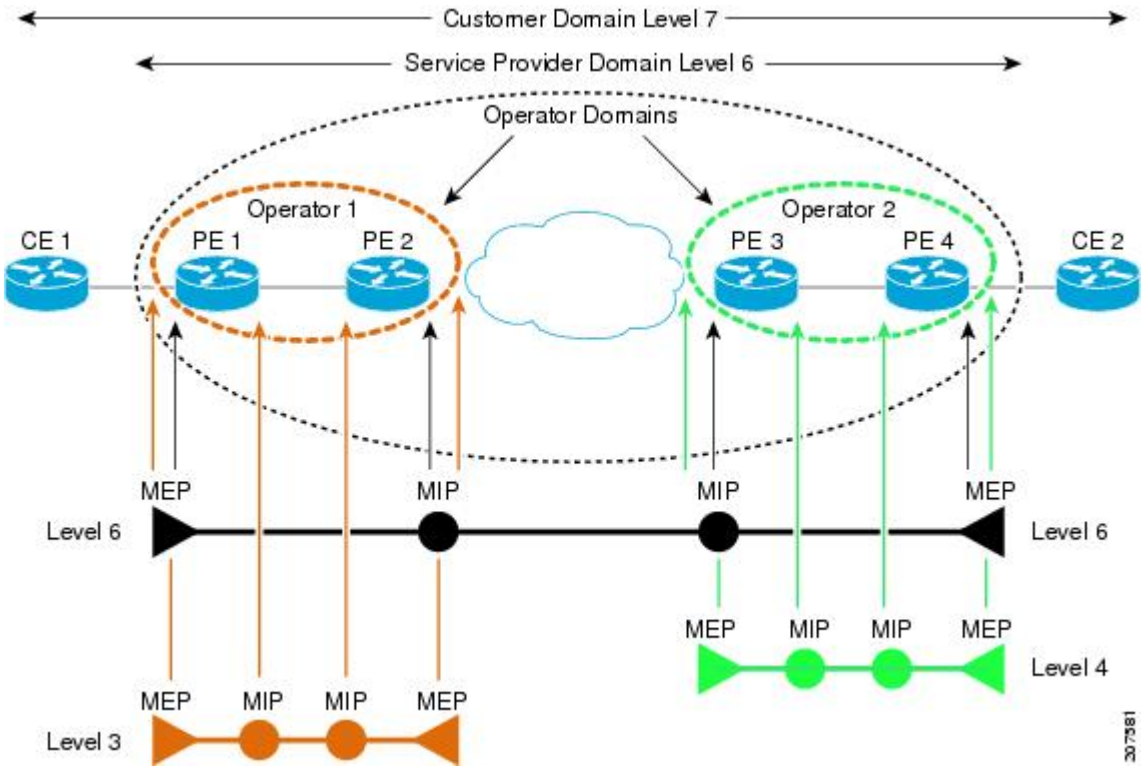
- The customer can use CFM between their CE devices, to verify and manage connectivity across the whole network.
- The service provider can use CFM between their PE devices, to verify and manage the services they are providing.
- Each operator can use CFM within their operator network, to verify and 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. For more information about MEPs and MIPs, see the "Maintenance Points" section on page 71.

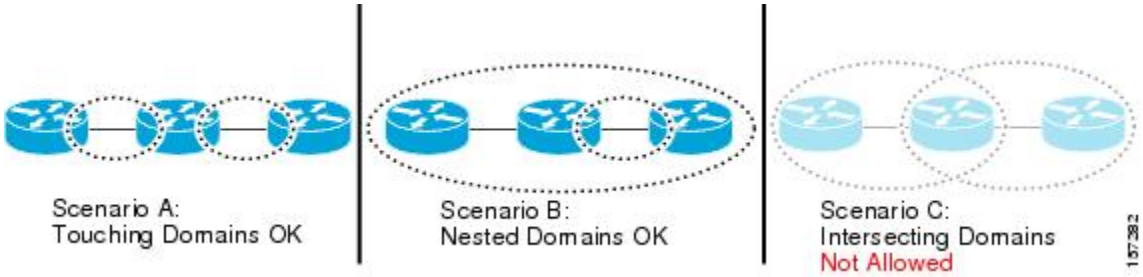
Figure 3: Different CFM Maintenance Domains Across a Network



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.

Figure 4: Data Supported CFM Maintenance Domain Structure



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 described in the "Maintenance Domains" section on page 69, and ensures that CFM frames for a particular domain cannot leak out beyond the boundary of the domain.

There are two types of MP:

- 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.
- Maintenance Intermediate Points (MIPs)—Created in the middle of the domain. Unlike MEPS, MIPs do allow CFM frames at their own level to be forwarded.

MIP Creation

Unlike MEPs, MIPs are not explicitly configured on each interface. MIPs are created automatically according to the algorithm specified in the CFM 802.1ag standard. The algorithm, in brief, operates as follows for each interface:

- The cross-connect for the interface is found, and all services associated with that cross-connect are considered for MIP auto-creation.
- The level of the highest-level MEP on the interface is found. From among the services considered above, the service in the domain with the lowest level that is higher than the highest MEP level is selected. If there are no MEPs on the interface, the service in the domain with the lowest level is selected.

- The MIP auto-creation configuration (**mip auto-create** command) for the selected service is examined to determine whether a MIP should be created.

Note Configuring a MIP auto-creation policy for a service does not guarantee that a MIP will automatically be created for that service. The policy is only considered if that service is selected by the algorithm first.

MEP and CFM Processing Overview

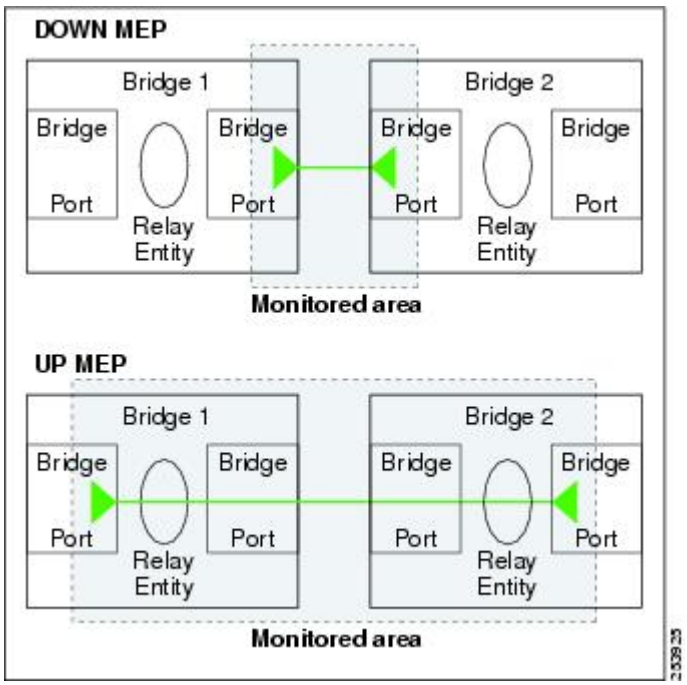
The boundary of a domain is an interface, rather than a bridge or host. Therefore, MEPs can be sub-divided into two categories:

- 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.

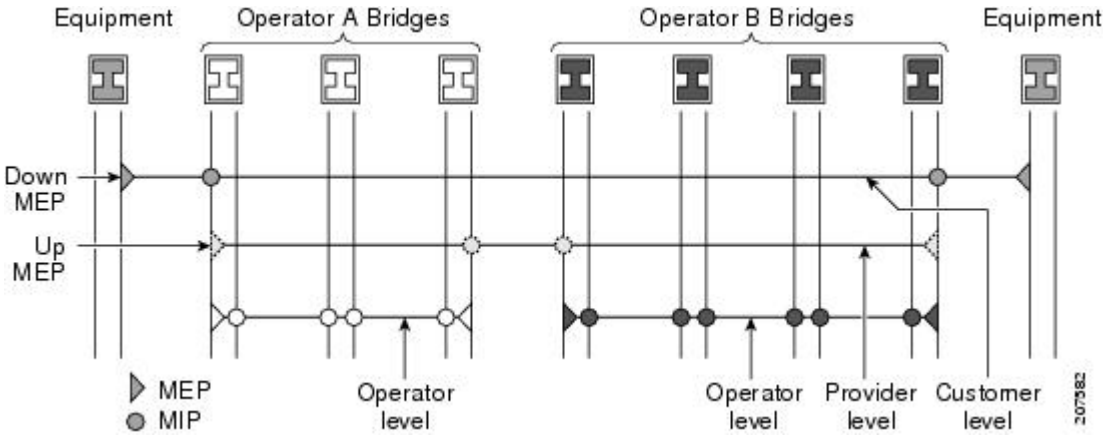
This figure illustrates the monitored areas for Down and Up MEPs.

Figure 5: 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 always corresponds with a MEP or MIP at a higher level. In addition, only a single MIP is allowed on any interface—this is generally created in the lowest domain that exists at the interface and that does not have a MEP.

Figure 6: Data CFM Maintenance Points at Different Levels



MIPs and 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) or routed (Layer 3) 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.

MIPs also continue to receive CFM frames at their level if the interface is STP blocked, and can respond to any received frames. However, MIPs do not allow CFM frames at the level of the MIP to be forwarded if the interface is blocked.

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.

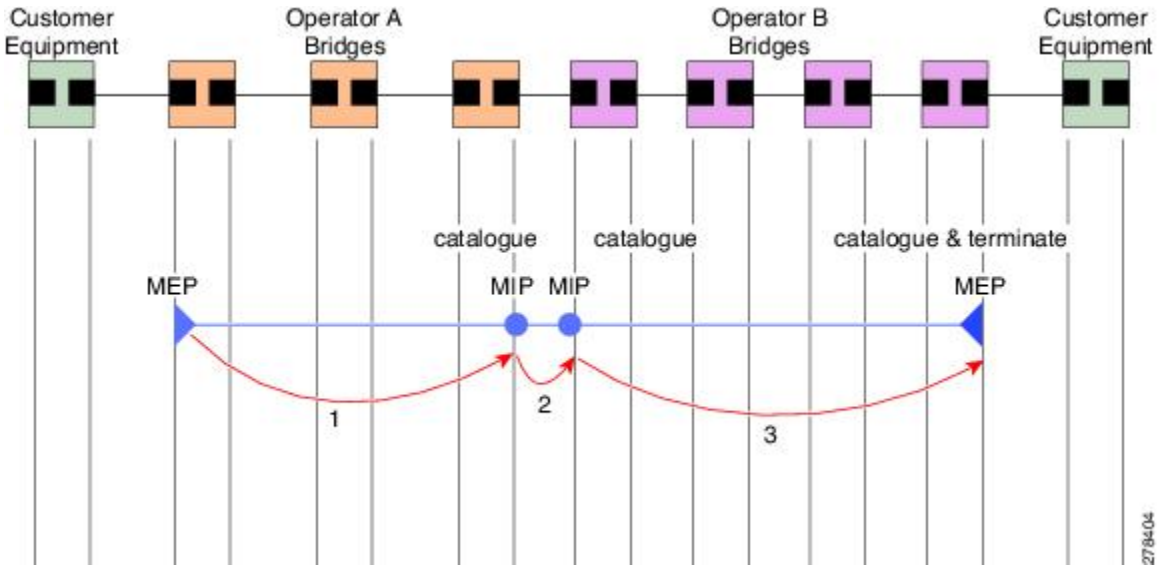
This section describes the following CFM messages:

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.

Figure 7: Continuity Check Message Flow



All the MEPs in a service must transmit CCMs at the same interval. IEEE 802.1ag defines 7 possible intervals that can be used:

- 10ms

- 100ms
- 1s
- 10s
- 1 minute
- 10 minutes

A MEP detects a loss of connectivity with one of its peer MEPs when some number of CCMs have been 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.

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.
- A configured numeric identifier for the MEP (the MEP ID). Each MEP in the service must be configured with a different MEP ID.
- A sequence number.
- 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.
- 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 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 an 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.

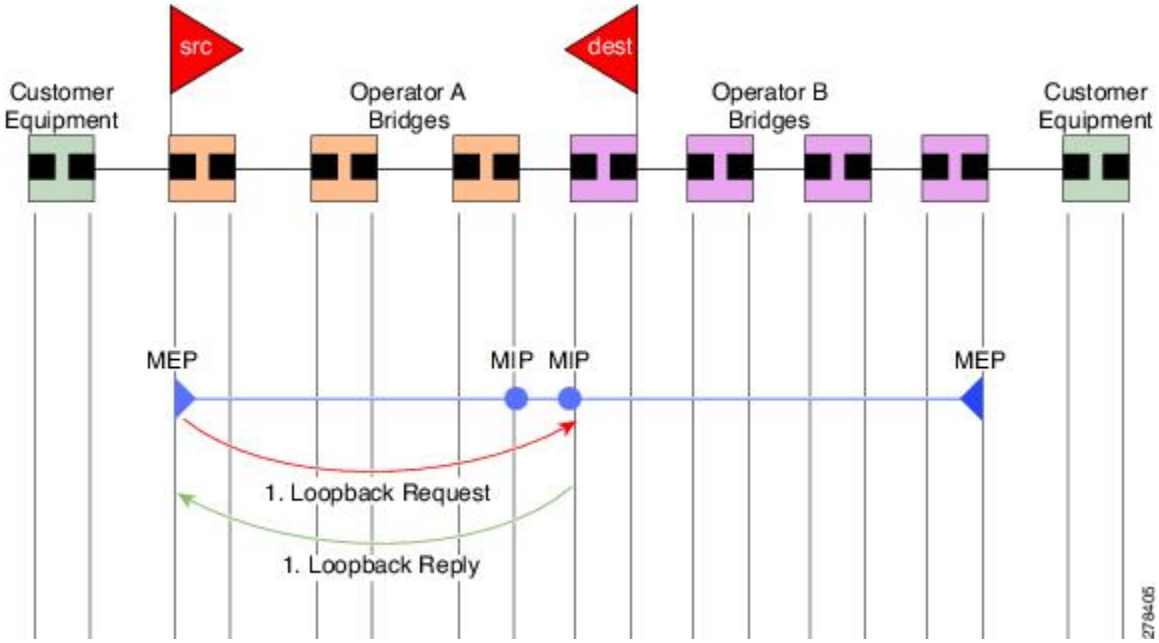
Out-of-sequence CCMs can also be detected by monitoring the sequence number in the received CCMs from each peer MEP. However, this is not considered a CCM defect.

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 MIP.

Figure 8: 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.

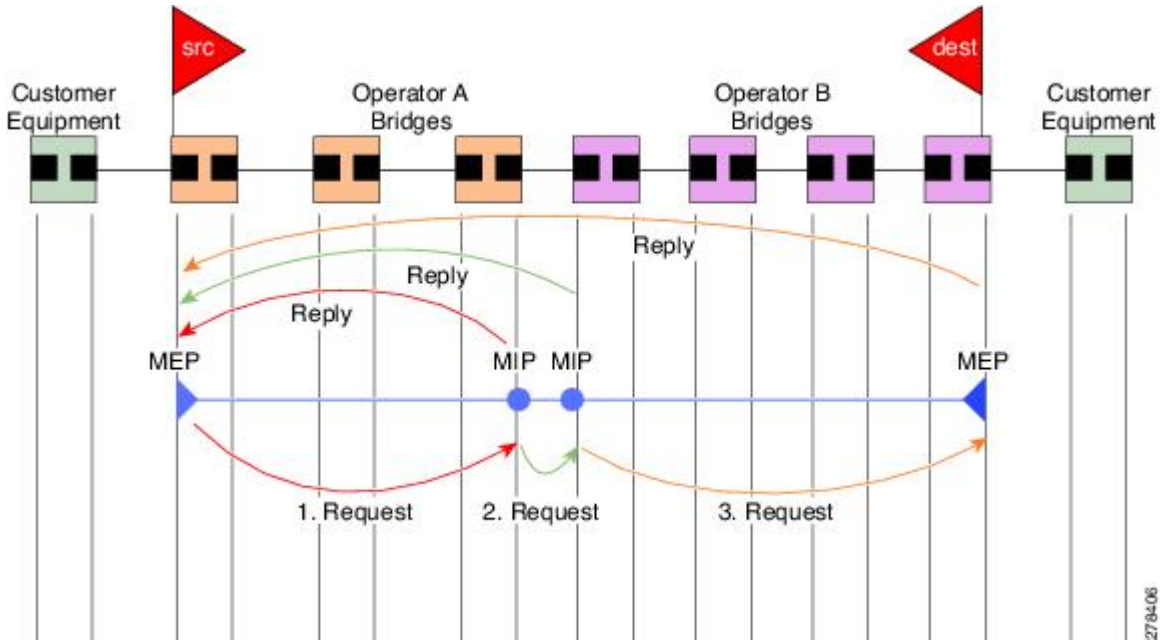
Note The Ethernet CFM loopback function should not be confused with the remote loopback functionality in Ethernet Link OAM. CFM loopback is used to test connectivity with a remote MP, and only the CFM LBM packets are reflected back, but Ethernet Link OAM remote loopback is used to test a link by taking it out of normal service and putting it into a mode where it reflects back all packets.

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 LTM. Each hop where there is a maintenance point sends an 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 MIPs.

Figure 9: 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.

Exploratory Linktrace (Cisco)

Exploratory Linktrace is a Cisco extension to the standard linktrace mechanism described above. It has two primary purposes:

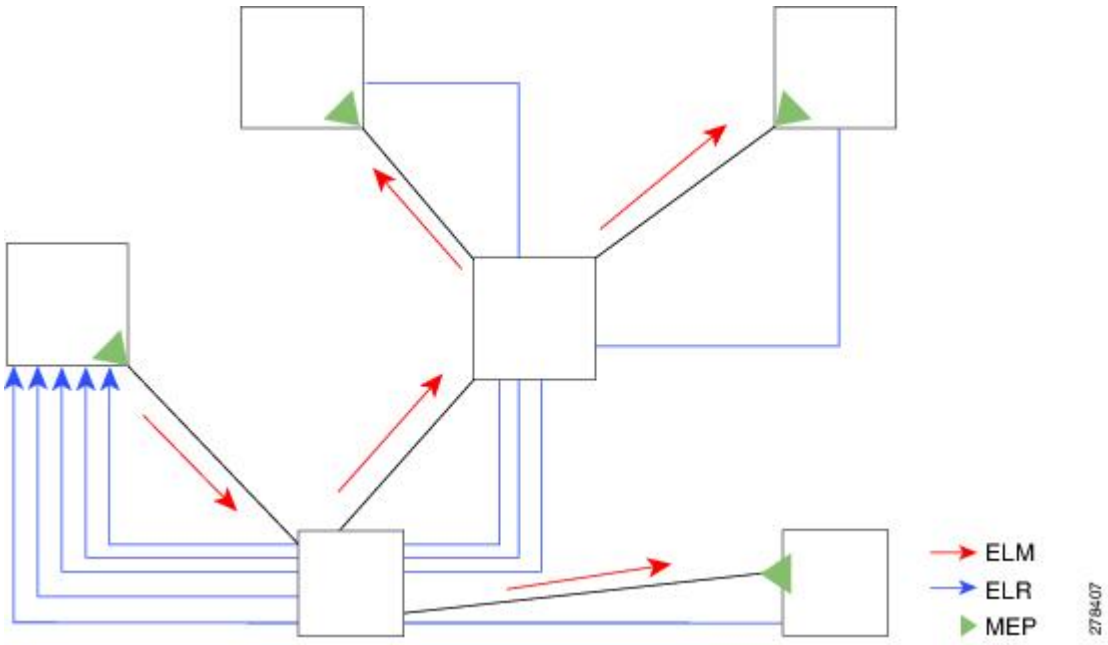
- Provide a mechanism to locate faults in cases where standard linktrace does not work, such as when a MAC address has never been seen previously in the network. For example, if a new MEP has been provisioned but is not working, standard linktrace does not help isolate a problem because no frames will ever have been received from the new MEP. Exploratory Linktrace overcomes this problem.
- Provide a mechanism to map the complete active network topology from a single node. This can only be done currently by examining the topology (for example, the STP blocking state) on each node in the network individually, and manually combining this information to create the overall active topology map. Exploratory linktrace allows this to be done automatically from a single node.

Exploratory Linktrace is implemented using the Vendor Specific Message (VSM) and Vendor Specific Reply (VSR) frames defined in ITU-T Y.1731. These allow vendor-specific extensions to be implemented without degrading interoperability. Exploratory Linktrace can safely be deployed in a network that includes other CFM implementations because those implementations will simply ignore the Exploratory Linktrace messages.

Exploratory Linktrace is initiated at the request of the administrator, and results in the local MEP sending a multicast Exploratory Linktrace message. Each MP in the network that receives the message sends an Exploratory Linktrace reply. MIPs that receive the message also forward it on. The initiating MEP uses all the replies to create a tree of the overall network topology.

This figure show an example of the Exploratory Linktrace message flow between MEPs.

Figure 10: Exploratory Linktrace Messages and Replies



To avoid overloading the originating MEP with replies in a large network, responding MPs delay sending their replies for a random amount of time, and that time increases as the size of the network increases.

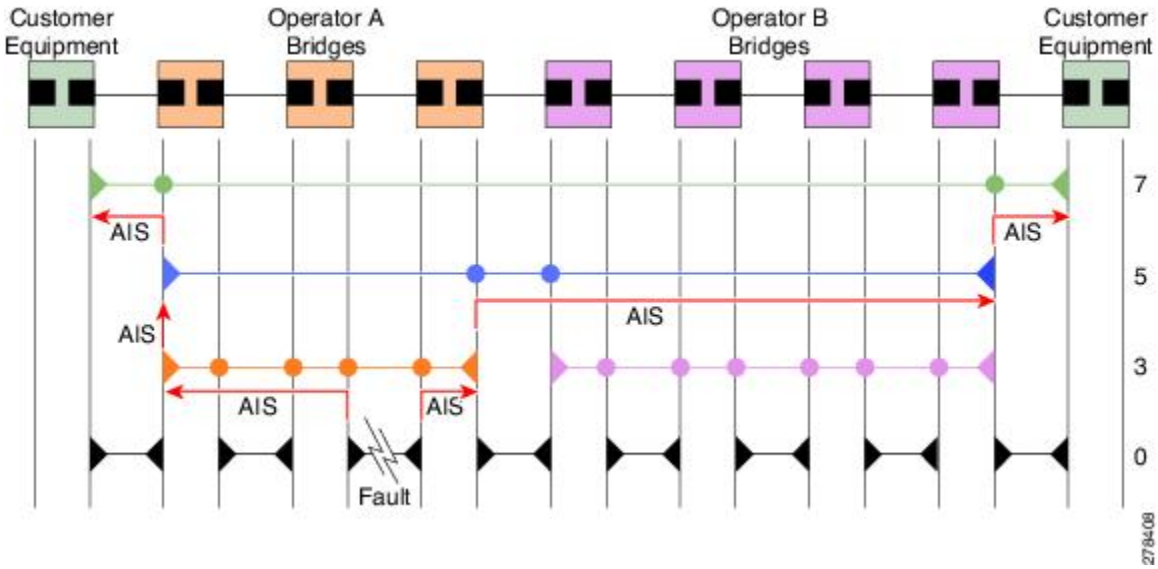
In a large network, there will be a corresponding large number of replies and the resulting topology map will be equally large. If only a part of the network is of interest, for example, because a problem has already been narrowed down to a small area, then the Exploratory Linktrace can be “directed” to start at a particular MP. Replies will thus only be received from MPs beyond that point in the network. The replies are still sent back to the originating MEP.

Alarm Indication Signal (ITU-T Y.1731)

Alarm Indication Signal (AIS) messages are used to rapidly notify MEPs when a fault is detected in the middle of a domain, in an event driven way. MEPs thereby learn of the fault much sooner than if they relied on detecting a loss of continuity, for example, failure to receive some number of consecutive CCMs.

Unlike all other CFM messages, AIS messages are injected into the middle of a domain, and sent outward toward the MEPs at the edge of the domain. Typically, AIS messages are injected by a MEP in a lower level domain. To put it another way, when a MEP sends AIS messages, they are sent in the opposite direction to other CFM messages sent by the MEP, and at a level above the MEP’s own level. The AIS messages are received by the MEPs in the higher level domain, not by the peer MEPs in the same domain as the MEP sending the AIS. When a MEP receives an AIS message, it may itself send another AIS message at an even higher level.

Figure 11: AIS Message Flow



AIS is only applicable in point-to-point networks. In multipoint networks with redundant paths, a failure at a low level does not necessarily result in a failure at a higher level, as the network may reconverge so as to route around the failed link.

AIS messages are typically sent by a MEP. However, AIS messages can also be sent when there is no MEP present, if a fault is detected in the underlying transport, such as if an interface goes down. In ITU-T Y.1731 these are referred to as server MEPs.

AIS messages are sent in response to a number of failure conditions:

- Detection of CCM defects, as described Continuity Check (IEEE 802.1ag and ITU-T Y.1731), on page 13.
- Loss of continuity.
- Receipt of AIS messages.
- Failure in the underlying transport, such as when an interface is down.

Received AIS messages can be used to detect and act on failures more quickly than waiting for a loss of continuity. They can also be used to suppress any failure action, on the basis that the failure has already been detected at a lower level and will be handled there. This is described in ITU-T Y.1731; however, the former is often more useful.

MEP Cross-Check

MEP cross-check supports configuration of a set of expected peer MEPs so that errors can be detected when any of the known MEPs are missing, or if any additional peer MEPs are detected that are not in the expected group.

The set of expected MEP IDs in the service is user-defined. Optionally, the corresponding MAC addresses can also be specified. CFM monitors the set of peer MEPs from which CCMs are being received. If no CCMs are ever received from one of the specified expected peer MEPs, or if a loss of continuity is detected, then a cross-check “missing” defect is detected. Similarly, if CCMs are received from a matching MEP ID but with

the wrong source MAC address, a cross-check “missing” defect is detected. If CCMs are subsequently received that match the expected MEP ID, and if specified, the expected MAC address, then the defect is cleared.

Note In NCS4K, CFM cross-check is mandatory for CFM offloaded session. Cross-check feature can be configured with or without mac address option. Cross-check is not mandatory for non-offloaded session.

If cross-check is configured and CCMs are received from a peer MEP with a MEP ID that is not expected, this is detected as a cross-check “unexpected” condition.

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.

EFD

Ethernet Fault Detection (EFD) is a mechanism that allows Ethernet OAM protocols, such as CFM, to control the “line protocol” state of an interface.

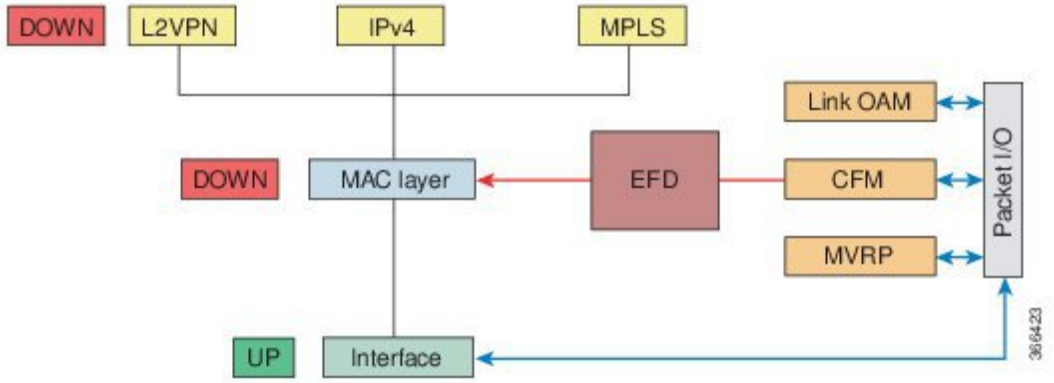
Unlike many other interface types, Ethernet interfaces do not have a line protocol, whose state is independent from that of the interface. For Ethernet interfaces, this role is handled by the physical-layer Ethernet protocol itself, and therefore if the interface is physically up, then it is available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

EFD changes this to allow CFM to act as the line protocol for Ethernet interfaces. This allows CFM to control the interface state so that if a CFM 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 any traffic flowing, 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 CFM frames continue to flow. This allows CFM to detect when the problem has been resolved, and thus bring the interface backup automatically.

This figure shows CFM 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 CFM detects there is no longer any error, it can signal to EFD and all protocols will once again go active.

Figure 12: CFM Error Detection and EFD Trigger



Flexible VLAN Tagging for CFM

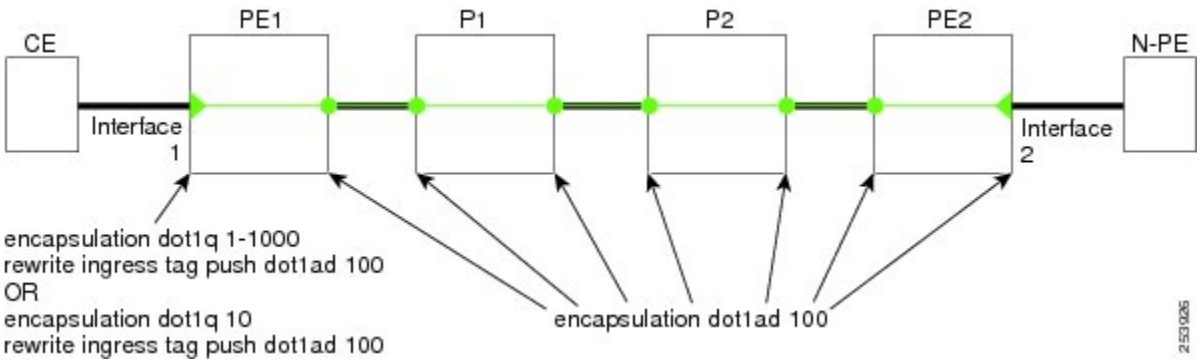
The Flexible VLAN Tagging for CFM feature ensures that CFM packets are sent with the right VLAN tags so that they are appropriately handled as a CFM packet by the remote device. When packets are received by an edge router, they are treated as either CFM packets or data packets, depending on the number of tags in the header. The system differentiates between CFM packets and data packets based on the number of tags in the packet, and forwards the packets to the appropriate paths based on the number of tags in the packet.

CFM frames are normally sent with the same VLAN tags as the corresponding customer data traffic on the interface, as defined by the configured encapsulation and tag rewrite operations. Likewise, received frames are treated as CFM frames if they have the correct number of tags as defined by the configured encapsulation and tag rewrite configuration, and are treated as data frames (that is, they are forwarded transparently) if they have more than this number of tags.

In most cases, this behavior is as desired, since the CFM frames are then treated in exactly the same way as the data traffic flowing through the same service. However, in a scenario where multiple customer VLANs are multiplexed over a single multipoint provider service (for example, N:1 bundling), a different behavior might be desirable.

This figure shows an example of a network with multiple VLANS using CFM.

Figure 13: Service Provider Network With Multiple VLANs and CFM



This figure shows a provider's access network, where the S-VLAN tag is used as the service delimiter. PE1 faces the customer, and PE2 is at the edge of the access network facing the core. N:1 bundling is used, so the interface encapsulation matches a range of C-VLAN tags. This could potentially be the full range, resulting in all:1 bundling. There is also a use case where only a single C-VLAN is matched, but the S-VLAN is nevertheless used as the service delimiter—this is more in keeping with the IEEE model, but limits the provider to 4094 services.

CFM is used in this network with a MEP at each end of the access network, and MIPs on the boxes within the network (if it is native Ethernet). In the normal case, CFM frames are sent by the up MEP on PE1 with two VLAN tags, matching the customer data traffic. This means that at the core interfaces and at the MEP on PE2, the CFM frames are forwarded as if they were customer data traffic, since these interfaces match only on the S-VLAN tag. So, the CFM frames sent by the MEP on PE1 are not seen by any of the other MPs.

Flexible VLAN tagging changes the encapsulation for CFM frames that are sent and received at Up MEPs. Flexible VLAN tagging allows the frames to be sent from the MEP on PE1 with just the S-VLAN tag that represents the provider service. If this is done, the core interfaces will treat the frames as CFM frames and they will be seen by the MIPs and by the MEP on PE2. Likewise, the MEP on PE1 should handle received frames with only one tag, as this is what it will receive from the MEP on PE2.

To ensure that CFM packets from Up MEPs are routed to the appropriate paths successfully, tags may be set to a specific number in a domain service, using the **tags** command. Currently, tags can only be set to one (1).

CFM Scale Details

The following table has CFM scale details:

Packet Type	Scale
CCM	2000 per LC, 8000 per system
AIS	2000 per LC, 8000 per system
SLM	2000 per LC, 8000 per system
Two-way DM	2000 per LC, 8000 per system

The scale numbers indicated in the above table are applicable for single chassis and multi chassis systems.

Ethernet SLA

Customers require their service providers to conform to a Service Level Agreement (SLA). Consequently, service providers must be able to monitor the performance characteristics of their networks. Similarly, customers also want to monitor the performance characteristics of their networks. Cisco provides Y.1731 performance monitoring using the Cisco Ethernet SLA feature.

The Cisco Ethernet SLA feature provides the architecture to monitor a network at Layer 2. This architecture provides functions such as collecting, storing, displaying, and analyzing SLA statistics. These SLA statistics can be stored and displayed in various ways, so that statistical analysis can be performed.

Ethernet SLA provides the framework for performing the following major functions of performance monitoring:

- Sending probes consisting of one or more packets to measure performance.

  Ethernet SLA provides a flexible mechanism for sending SLA probes to measure performance. Probes can consist of either CFM loopback, CFM loss measurement packets, or CFM delay measurement packets.

Options are available to modify how often the packets are sent, and to specify the attributes of the probe packets such as the size and priority.

- Scheduling of operations consisting of periodic probes.

  A flexible mechanism is provided by Ethernet SLA to specify how often each probe should be executed, how long it should last, and when the first probe should start. Probes can be scheduled to run back-to-back to provide continuous measurements, or at a defined interval ranging from once a minute to once a week.

- Collecting and storing results.

  Ethernet SLA provides flexibility to specify which performance parameters should be collected and stored for each measurement probe. Performance parameters include frame delay and jitter (inter-frame delay variation). For each performance parameter, either each individual result can be stored, or the results can be aggregated by storing a counter of the number of results that fall within a particular range. A configurable amount of historical data can also be stored as well as the latest results.

- Analyzing and displaying results.

  Ethernet SLA performs some basic statistical analysis on the collected results, such as calculating the minimum, maximum, mean and standard deviation. It also records whether any of the probe packets were lost or misordered, or if there is any reason why the results may not be a true reflection of the performance (for example if a big jump in the local time-of-day clock was detected during the time when the measurements were being made).

Ethernet SLA Measurement Packet

An Ethernet SLA *measurement packet* is a single protocol message and corresponding reply that is sent on the network for the purpose of making SLA measurements. These types of measurement packet are supported:

- CFM Delay Measurement (Y.1731 DMM/DMR packets)—CFM delay measurement packets contain timestamps within the packet data that can be used for accurate measurement of frame delay and jitter. These packets can be used to measure round-trip statistics; however, the size of the DMM/DMR packets cannot be modified.



Note Delay measurement can be performed only after the PTP is configured and the clocks are in sync. See Understanding PTP, on page 1.

- CFM loopback (LBM/LBR)—CFM loopback packets are less accurate, but can be used if the peer device does not support DMM/DMR packets. Only round-trip statistics can be measured because these packets do not contain timestamps. However, loopback packets can be padded, so measurements can be made using frames of a specific size.

- CFM Synthetic Loss Measurement (Y.1731 SLM/SLR packets)—SLM packets contain two sequence numbers; one written by the initiator into the SLM and copied by the responder into the SLR, and the other allocated by the responder and written into the SLR. These are refered to as the source-to-destination (sd) sequence number and the destination-to-source (ds) sequence number respectively.

Ethernet SLA Sample

A *sample* is a single result—a number—that relates to a given statistic. For some statistics such as round-trip delay, a sample can be measured using a single measurement packet. For other statistics such as jitter, obtaining a sample requires two measurement packets.

Ethernet SLA Probe

A *probe* is a sequence of measurement packets used to gather SLA samples for a specific set of statistics. The measurement packets in a probe are of a specific type (for example, CFM delay measurement or CFM loopback) and have specific attributes, such as the frame size and priority.

Ethernet SLA Burst

Within a probe, measurement packets can either be sent individually, or in bursts. A *burst* contains two or more packets sent within a short interval. Each burst can last up to one minute, and bursts can follow each other immediately to provide continuous measurement within the probe.

For statistics that require two measurement packets for each sample, samples are only calculated based on measurement packets in the same burst. For all statistics, it is more efficient to use bursts than to send individual packets.

Ethernet SLA Schedule

An Ethernet SLA *schedule* describes how often probes are sent, how long each probe lasts, and at what time the first probe starts.

Ethernet SLA Bucket

For a particular statistic, a *bucket* is a collection of results that were gathered during a particular period of time. All of the samples for measurements that were initiated during the period of time represented by a bucket are stored in that bucket. Buckets allow results from different periods of time to be compared (for example, peak traffic to off-peak traffic).

By default, a separate bucket is created for each probe; that is, the bucket represents the period of time starting at the same time as the probe started, and continuing for the duration of the probe. The bucket will therefore contain all the results relating to measurements made by that probe.

Ethernet SLA Operation

An *operation* is an instance of a given operation profile that is actively collecting performance data. Operation instances are created by associating an operation profile with a given source (an interface and MEP) and with a given destination (a MEP ID or MAC address). Operation instances exist for as long as the configuration is applied, and they run for an indefinite duration on an ongoing basis.

Ethernet SLA On-Demand Operation

An *on-demand operation* is a method of Ethernet SLA operation that can be run on an as-needed basis for a specific and finite period of time. This can be useful in situations such as when you are starting a new service or modifying the parameters for a service to verify the impact of the changes, or if you want to run a more detailed probe when a problem is detected by an ongoing scheduled operation.

On-demand operations do not use profiles and have a finite duration. The statistics that are collected are discarded after a finite time after the operation completes (two weeks), or when you manually clear them. On-demand operations do not persist across a card reload.

Configuring SLA Operation

This section describes how to configure an ongoing SLA operation on a MEP using an SLA profile.

Procedure

Step 1 **interface** *type* **R/S/I/P**

Example:

```
RP/0/RP0:router(config)# interface gigabitethernet 0/1/0/1
```

Enters the interface configuration mode.

Step 2 **ethernet cfm**

Example:

```
RP/0/RP0:router(config-if)# ethernet cfm
```

Enters the CFM configuration mode.

Step 3 **mep domain** *domain_name* **service** *service_name* **mep-id** *number*

Example:

```
RP/0/RP0:router(config-if-cfm)# mep domain d1 service Sv1 mep-id 2
```

Creates a MEP on an interface and enters interface CFM MEP configuration mode.

Step 4 **sla operation profile** *profile_name* **target** s{ **mep-id** *id* | **mac-address** *address*}

Example:

```
RP/0/RP0:router(config-if-cfm-mep)# sla operation profile p1 target mac-address 01:23:45:67
```

Creates an operation instance from a MEP to a specified destination.

Step 5 **commit or end**

Saves the configuration changes; when you issue the **end** command, the system prompts you to commit the changes.

Configuring SLA Probe Profile

To configure SLA probe parameters in a profile, perform these steps beginning in SLA profile configuration mode.

Procedure

Step 1 **probe**

Example:

```
RP/0/RP0:router(config-sla-profile) # probe
```

Enters the SLA profile probe configration mode.

Step 2 **send** { **burst** | **packet** } { **every** *number* { **seconds** | **minutes** | **hours** } | **once** } } **packet count** *packets* **interval** *number* { **seconds** | **milliseconds** }

Example:

```
RP/0/RP0:router(config-sla-prof-pb)# send burst every 60 seconds packet count 100 interval
 100 milliseconds
```

Sets the parameters for burst or packet.

Step 3 **packet size** *bytes* [**test-pattern** { **hex 0xHHHHHHHH** | **pseudo-random** }]

Example:

```
RP/0/RP0:router(config-sla-prof-pb)# packet size 9000
```

Configures the minimum size (in bytes) for outgoing probe packets, including padding when necessary. Use the test pattern keyword to specify a hexadecimal string to use as the padding characters, or a pseudo-random bit sequence. The default padding is 0's. The packet size can be configured for SLM, loopback, and DMM/R probes.

Step 4 **priority**

Example:

```
RP/0/RP0:router(config-sla-prof-pb)# priority 7
```

Configures the priority of outgoing SLA probe packets.

Step 5 **synthetic loss calculation packets** *number*

Example:

```
RP/0/RP0:router(config-sla-prof-pb)# synthetic loss calculation packets 25
```

Configures the number of packets that must be used to make each frame loss ratio calculation in the case of synthetic loss measurements. This item can only be configured for packet types that support synthetic loss measurement.

Step 6 **commit or end**

Saves the configuration. When you use the **end** command, the system prompts the user to commit the changes.

Configuring SLA Operation Profile

This task has details about configuring an SLA operation profile. You can configure only up to hundred SLA operation profiles.

Procedure

Step 1 **configure**

Example:

```
RP/0/RP0/CPU0:router# config
```

Enters the global configuration mode.

Step 2 **ethernet sla**

Example:

```
RP/0/RP0/CPU0:router (config)# ethernet sla
```

Enters the SLA configuration mode.

Step 3 **profile** *profile-name* **type** { **cfm-delay-measurement** | **cfm-loopback** | **cfm-synthetic-loss-measurement** }

Example:

```
RP/0/RP0/CPU0:router(config-sla)# profile  profile1 type cfm-synthetic-loss-measurement
```

Creates an SLA operation profile and enters the SLA profile configuration mode.

Step 4 **commit or end**

Saves the configuration changes; when you issue the **end** command, the system prompts you to commit the changes.

Configuring SLA Statisics Profile

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

To configure SLA statistics measurement in a profile, perform these steps beginning in SLA profile configuration mode.

Procedure

Step 1 **statistics measure** { **round-trip-delay** | **round-trip-jitter** }

Example:

```
RP/0/RP0:router(config-sla-prof)# statistics measure round-trip-delay
```

Enables the collection of SLA statistics, and enters SLA profile statistics configuration mode.

Step 2 **aggregate** { **bins** *count* **width** *width* | **none** }

Example:

```
RP/0/RP0:router(config-sla-prof-stat-cfg)# aggregate bins 100 width 10000
```

Configures 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, the default is aggregation disabled.

Step 3 **buckets size** *number* { **probes** }

Example:

```
RP/0/RP0:router(config-sla-prof-stat-cfg)# buckets size 100 probes
```

Configures the size of the buckets in which statistics are collected.

Step 4 **buckets archive** *number*

Example:

```
RP/0/RP0:router(config-sla-prof-stat-cfg)# buckets archive 50
```

Configures the number of buckets to store in memory.

Step 5 **end or commit**

Saves the configuration changes; if you issue the **end** command, the system will prompt you to commit.

Requesting On-Demand Ethernet SLA for CFM Delay Measurement

This task has details about requesting an on-demand ethernet SLA operation for CFM delay measurement.

Procedure

ethernet sla on-demand operation type cfm-delay-measurement probe [**prority** *number*] [**send** { **packet** | **burst** *imterval* }] **domain** *domain name* **source interface** *type* **R/S/I/P target** { **mac-address** *address* | **mep-id***id* } [**statistics measure** { **round-trip-delay** | **round-trip-jitter**}] [**schedule** { **now** | **at** *time*] [**for duration** { **seconds** | **minutes** | **hours**}]

Example:

```
RP/0/RP0:router # ethernet sla on-demand operation type cfm-delay-measurement probe domain
 D1 source interface TenGigE 0/6/1/0 target mep-id 100
```

Configures an on-demand Ethernet SLA operation for CFM delay measurement.

Note
This command is in EXEC mode.

Configuring On-Demand Ethernet SLA for CFM Synthetic Loss Measurement

This task has details about configuring an on-demand ethernet SLA operation for CFM synthetic loss measurement.

Procedure

ethernet sla on-demand operation type cfm-synthetic-loss-measurement probe [**prority** *number*] [**send** { **packet** | **burst** *imterval* }] **domain** *domain name* **source interface** *type* **R/S/I/P target** { **mac-address** *address* | **mep-id***id* } [**synthetic loss calculation packets** *number*] [**statistics measure** { **round-trip-loss-ds** | **round-trip-loss-sd**}] [**schedule** { **now** | **at** *time*] [**for duration** { **seconds** | **minutes** | **hours**}]

Example:

```
RP/0/RP0:router (config)# ethernet sla on-demand operation type cfm-synthetic-loss-measurement
 probe domain D1 source interface TenGigE 0/6/1/0 target mac-address 2.3.4
```

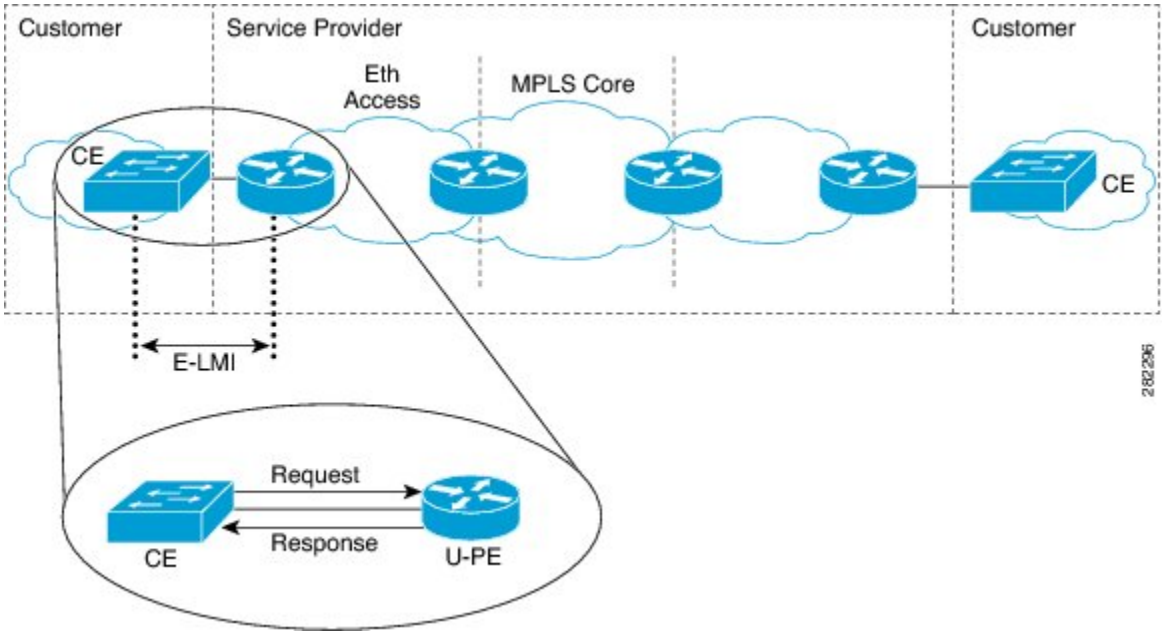
Configures an on-demand Ethernet SLA operation for CFM synthetic measurement.

Note
This command is in EXEC mode.

Ethernet LMI

E-LMI runs on the link between the customer-edge (CE) device and the provider-edge (PE) device, or User Network Interface (UNI), and provides a way for the CE device to auto-configure or monitor the services offered by the PE device (see this figure).

Figure 14: E-LMI Communication on CE-to-PE Link



E-LMI is an asymmetric protocol whose basic operation involves the User-facing PE (uPE) device providing connectivity status and configuration parameters to the CE using STATUS messages in response to STATUS ENQUIRY messages sent by the CE to the uPE.

E-LMI Messaging

The E-LMI protocol as defined by the MEF 16 standard, defines the use of only two message types—STATUS ENQUIRY and STATUS.

These E-LMI messages consist of required and optional fields called information elements, and all information elements are associated with assigned identifiers. All messages contain the Protocol Version, Message Type, and Report Type information elements, followed by optional information elements and sub-information elements.

E-LMI messages are encapsulated in 46- to 1500-byte Ethernet frames, which are based on the IEEE 802.3 untagged MAC-frame format. E-LMI frames consist of the following fields:

- Destination address (6 bytes)—Uses a standard MAC address of 01:80:C2:00:00:07.
- Source address (6 bytes)—MAC address of the sending device or port.
- E-LMI Ethertype (2 bytes)—Uses 88-EE.
- E-LMI PDU (46–1500 bytes)—Data plus 0x00 padding as needed to fulfill minimum 46-byte length.
- CRC (4 bytes)—Cyclic Redundancy Check for error detection.

Cisco-Proprietary Remote UNI Details Information Element

The E-LMI MEF 16 specification does not define a way to send proprietary information.

To provide additional information within the E-LMI protocol, the Cisco IOS XR software implements a Cisco-proprietary information element called Remote UNI Details to send information to the CE about remote UNI names and states. This information element implements what is currently an unused identifier from the E-LMI MEF 16 specification.

To ensure compatibility for future implementations of E-LMI should this identifier ever be implemented in the standard protocol, or for another reason, you can disable transmission of the Remote UNI information element using the **extension remote-uni disable** command.

E-LMI Operation

The basic operation of E-LMI consists of a CE device sending periodic STATUS ENQUIRY messages to the PE device, followed by mandatory STATUS message responses by the PE device that contain the requested information. Sequence numbers are used to correlate STATUS ENQUIRY and STATUS messages between the CE and PE.

The CE sends the following two forms of STATUS ENQUIRY messages called Report Types:

- E-LMI Check—Verifies a Data Instance (DI) number with the PE to confirm that the CE has the latest E-LMI information.
- Full Status—Requests information from the PE about the UNI and all EVCs.

The CE device uses a polling timer to track sending of STATUS ENQUIRY messages, while the PE device can optionally use a Polling Verification Timer (PVT), which specifies the allowable time between transmission of the PE's STATUS message and receipt of a STATUS ENQUIRY from the CE device before recording an error.

In addition to the periodic STATUS ENQUIRY/STATUS message sequence for the exchange of E-LMI information, the PE device also can send asynchronous STATUS messages to the CE device to communicate changes in EVC status as soon as they occur and without any prompt by the CE device to send that information.

Both the CE and PE devices use a status counter (N393) to determine the local operational status of E-LMI by tracking consecutive errors received before declaring a change in E-LMI protocol status.

Supported E-LMI PE Functions

The Cisco NCS 4000 Series Router serves as the PE device for E-LMI on a MEN, and supports the following PE functions:

- Supports the E-LMI protocol on Ethernet physical interfaces that are configured with Layer 2 subinterfaces as Ethernet Flow Points (EFPs), which serve as the EVCs about which the physical interface reports status to the CE. The Cisco IOS XR software does not support a specific manageability context for an Ethernet Virtual Connection (EVC).



Note For E-LMI on the Cisco NCS 4000 Series Router, the term EVC in this documentation refers to a Layer 2 subinterface/EFP.

- Provides the ability to configure the following E-LMI options defined in the MEF 16 specification:
  - T392 Polling Verification Timer (PVT)
  - N393 Status Counter
- Sends notification of the addition and deletion of an EVC.
- Sends notification of the availability (active) or unavailability (inactive, partially active) status of a configured EVC.
- Sends notification of the local UNI name.
- Sends notification of remote UNI names and states using the Cisco-proprietary Remote UNI Details information element, and the ability to disable the Cisco-proprietary Remote UNI information element.
- Sends information about UNI and EVC attributes to the CE (to allow the CE to auto-configure these attributes), including:
  - CE-VLAN to EVC Map
  - CE-VLAN Map Type (Bundling, All-to-one Bundling, Service Multiplexing)
  - Service Type (point-to-point or multipoint)
- Uses CFM Up MEPs to retrieve the EVC state, EVC Service Type, and remote UNI details.
- Provides the ability to retrieve the per-interface operational state of the protocol (including all the information currently being communicated by the protocol to the CE) using the command-line interface (CLI) or Extensible Markup Language (XML) interface.
- Supports up to 80 E-LMI sessions per linecard (one per physical interface).
- Supports up to 32000 EVCs total per linecard for all physical interfaces enabled for E-LMI.

How to Configure Ethernet OAM

This section provides these configuration procedures:

Configuring Ethernet Link 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

Step 1 **configure**

Example:

```
RP/0/# configure terminal
```

Enters global configuration mode.

Step 2 **ethernet oam profile** *profile-name*

Example:

```
RP/0/(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/(config-eoam)# link-monitor
```

Enters the Ethernet OAM link monitor configuration mode.

Step 4 **symbol-period window** *window*

Example:

```
RP/0/(config-eoam-lm)# symbol-period window 60000
```

(Optional) Configures the window size (in milliseconds) for an Ethernet OAM symbol-period error event.

The range is 1000 to 60000.

The default value is 1000.

Step 5 **symbol-period threshold low** *threshold* **high** *threshold*

Example:

```
RP/0/(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 **frame window** *window*

Example:

```
RP/0/(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 7 **frame threshold low** *threshold* **high** *threshold*

Example:

```
RP/0/(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 8 **frame-period window** *window*

Example:

```
RP/0/(config-eoam-lm)# frame-period window 60000
```

(Optional) Configures the window size (in milliseconds) for an Ethernet OAM frame-period error event.

The range is from 100 to 60000.

The default value is 1000.

Step 9 **frame-period threshold low** *threshold* **high** *threshold*

Example:

```
RP/0/(config-eoam-lm)# frame threshold low 10000000 high 60000000
```

(Optional) Configures the thresholds (in frames) that trigger an Ethernet OAM frame-period error event. The high threshold is optional and is configurable only in conjunction with the low threshold.

The range is 0 to 1000000.

The default low threshold is 60000.

Step 10 **frame-seconds window** *window*

Example:

```
RP/0/(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.

Step 11 **frame-seconds threshold low** *threshold* **high** *threshold*

Example:

```
RP/0/(config-eoam-lm)# frame-seconds threshold 3 threshold 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 12 **exit**

Example:

```
RP/0/(config-eoam-lm)# exit
```

Exits back to Ethernet OAM mode.

Step 13 **mib-retrieval**

Example:

```
RP/0/(config-eoam)# mib-retrieval
```

Enables MIB retrieval in an Ethernet OAM profile or on an Ethernet OAM interface.

Step 14 **connection timeout** <*timeout*>

Example:

```
RP/0/(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 15 **hello-interval** {**100ms**|**1s**}

Example:

```
RP/0/(config-eoam)# hello-interval 100ms
```

Configures the time interval between hello packets for an Ethernet OAM session. The default is 1 second (**1s**).

Step 16 **mode** {**active**|**passive**}

Example:

```
RP/0/(config-eoam)# mode passive
```

Configures the Ethernet OAM mode. The default is active.

Step 17 **require-remote mode** {**active**|**passive**}

Example:

```
RP/0/(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 18 **require-remote link-monitoring**

Example:

```
RP/0/(config-eoam)# require-remote link-monitoring
```

Requires that link-monitoring is configured on the remote end before the OAM session becomes active.

Step 19 **require-remote mib-retrieval**

Example:

```
RP/0/(config-eoam)# require-remote mib-retrieval
```

Requires that MIB-retrieval is configured on the remote end before the OAM session becomes active.

Step 20 **action capabilities-conflict** {**disable | efd | error-disable-interface**}

Example:

```
RP/0/(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/(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/(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/(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/(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 mode to override the profile setting and take no action at the interface when the event occurs.

Step 25 **action remote-loopback disable**

Example:

```
RP/0/(config-eoam)# action remote-loopback disable
```

Specifies that no action is taken on an interface when a remote-loopback 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 26 **action session-down** {**disable** | **efd** | **error-disable-interface**}

Example:

```
RP/0/(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 27 **action session-up disable**

Example:

```
RP/0/(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 28 **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 29 **action wiring-conflict** {**disable | efd | log**}

Example:

```
RP/0/(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 30 **uni-directional link-fault detection**

Example:

```
RP/0/(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.

Step 31 **commit**

Example:

```
RP/0/(config-if)# commit
```

Saves the configuration changes to the running configuration file and remains within the configuration session.

Step 32 **end**

Example:

```
RP/0/(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

Step 1 **configure**

Example:

```
RP/0/# configure terminal
```

Enters global configuration mode.

Step 2 **interface** [**FastEthernet** | **HundredGigE**| **TenGigE**] *interface-path-id*

Example:

```
RP/0/(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/(config-if)# ethernet oam
```

Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.

Step 4 **profile** *profile-name*

Example:

```
RP/0/(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/(config-if)# commit
```

Saves the configuration changes to the running configuration file and remains within the configuration session.

Step 6 **end**

Example:

```
RP/0/(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 on page 107.

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

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure terminal
```

Enters global configuration mode.

Step 2 **interface** [TenGigE | HundredGigE] *interface-path-id*

Example:

```
RP/0/(config)# interface TenGigE 0/1/0/0
```

Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port*.

Step 3 **ethernet oam**

Example:

```
RP/0/(config-if)# ethernet oam
```

Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.

Step 4 *interface-Ethernet-OAM-command*

Example:

```
RP/0/(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/(config-if)# commit
```

Saves the configuration changes to the running configuration file and remains within the configuration session.

Step 6 **end**

Example:

```
RP/0/(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/# show ethernet oam configuration
Thu Aug  5 22:07:06.870 DST
TenGigE0/4/0/0:
  Hello interval:                                      1s
  Link monitoring enabled:                              Y
  Remote loopback enabled:                              N
  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
Remote loopback action:                                  Log
Require remote mode:                                  Ignore
Require remote MIB retrieval:                              N
Require remote loopback support:                           N
Require remote link monitoring:                            N
```

Configuring Ethernet CFM

To configure Ethernet CFM, perform the following tasks:

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:

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **ethernet cfm**

Example:

```
RP/0/# ethernet cfm
```

Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.

Step 3 **domain** *domain-name* **level** *level-value* **id null**

Example:

```
RP/0/(config-cfm)# domain Domain_One level 1 id null
```

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.

Step 4 **service** *service-name* [**down-meps** | **xconnect**] **id** [**icc-based** *icc-string* | **number** *number*]

Example:

```
RP/0/(config-cfm-dmn)# service Bridge_Service down-meps number 10
```

Configures and associates a service with the domain and enters CFM domain service configuration mode.

The **id** sets the short MA name.

Step 5 **mep crosscheck**

Example:

```
RP/0/(config-cfm-dmn-svc)# mep crosscheck mep-id 10
```

Enters CFM MEP crosscheck configuration mode.

Step 6 **mep-id** *mep-id-number* [**mac-address** *mac-address*]

Example:

```
RP/0/(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/(config-cfm-xcheck)# commit
```

Saves configuration changes.

Configuring a CFM Maintenance Domain

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

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **ethernet cfm**

Example:

```
RP/0/(config)# ethernet cfm
```

Enters Ethernet Connectivity Fault Management (CFM) configuration mode.

Step 3 **domain** *domain-name* **level** *level-value* **id null**

Example:

```
RP/0/(config-cfm)# domain Domain_One level 1 id null
```

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.

Step 4 **traceroute cache hold-time** *minutes* **size** *entries*

Example:

```
RP/0/(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/(config-cfm-dmn)# commit
```

Saves configuration changes.

Configuring Services for a CFM Maintenance Domain

You can configure up to 32000 CFM services for a maintenance domain.

Before you begin

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

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **ethernet cfm**

Example:

```
RP/0/(config)# ethernet cfm
```

Enters Ethernet CFM configuration mode.

Step 3 **domain** *domain-name* **level** *level-value* **id null**

Example:

```
RP/0/(config-cfm)# domain Domain_One level 1 id null
```

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.

Step 4 **service** *service-name* [**down-meps** | **xconnect**] **id** [**icc-based** *icc-string* | **number** *number*]

Example:

```
RP/0/(config-cfm-dmn)# service Bridge_Service down-meps number 10
```

Configures and associates a service with the domain and enters CFM domain service configuration mode.

The **id** sets the short MA name.

Step 5 **end** or **commit**

Example:

```
RP/0/(config-cfm-dmn-svc)# commit
```

Saves configuration changes.

Enabling and Configuring Continuity Check for a CFM Service

It supports Continuity Check as defined in the IEEE 802.1ag specification, and supports CCMs intervals of 100 ms and longer. The overall packet rates for CCM messages are up to 16000 CCMs-per-second sent, and up to 16000 CCMs-per-second received, per card.

Note If Ethernet SLA is configured, the overall combined packet rate for CCMs and SLA frames is 16000 frames-per-second in each direction, per card.

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

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **ethernet cfm**

Example:

```
RP/0/(config)# ethernet cfm
```

Enters Ethernet Connectivity Fault Management (CFM) configuration mode.

Step 3 **domain** *domain-name* **level** *level-value* **id null**

Example:

```
RP/0/(config-cfm)# domain Domain_One level 1 id null
```

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.

Step 4 **service** *service-name* [**down-meps** | **xconnect**] **id** [**icc-based** *icc-string* | **number** *number*]

Example:

```
RP/0/(config-cfm-dmn)# service Bridge_Service down-meps number 10
```

Configures and associates a service with the domain and enters CFM domain service configuration mode.

The **id** sets the short MA name.

Step 5 **continuity-check interval** *time* [**loss-threshold** *threshold*]

Example:

```
RP/0/(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/(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/(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/(config-cfm-dmn-svc)# commit
```

Saves configuration changes.

Configuring Automatic MIP Creation for a CFM Service

To configure automatic MIP creation for a CFM service, complete the following steps:

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **ethernet cfm**

Example:

```
RP/0/# ethernet cfm
```

Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.

Step 3 **domain** *domain-name* **level** *level-value* **id null**

Example:

```
RP/0/(config-cfm)# domain Domain_One level 1 id null
```

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.

Step 4 **service** *service-name* [**down-meps** | **xconnect**] **id** [**icc-based** *icc-string* | **number** *number*]

Example:

```
RP/0/(config-cfm-dmn)# service Bridge_Service down-meps number 10
```

Configures and associates a service with the domain and enters CFM domain service configuration mode.

The **id** sets the short MA name.

Step 5 **mip auto-create** {**all** | **lower-mep-only**}

Example:

```
RP/0/(config-cfm-dmn-svc)# mip auto-create all
```

(Optional) Enables the automatic creation of MIPs in a bridge domain or xconnect.

Step 6 **end** or **commit**

Example:

```
RP/0/(config-cfm-dmn-svc)# commit
```

Saves configuration changes.

Configuring Other Options for a CFM Service

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

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **ethernet cfm**

Example:

```
RP/0/(config)# ethernet cfm
```

Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.

Step 3 **domain** *domain-name* **level** *level-value* **id null**

Example:

```
RP/0/(config-cfm)# domain Domain_One level 1 id null
```

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.

Step 4 **service** *service-name* [**down-meps** | **xconnect**] **id** [**icc-based** *icc-string* | **number** *number*]

Example:

```
RP/0/(config-cfm-dmn)# service Bridge_Service down-meps number 10
```

Configures and associates a service with the domain and enters CFM domain service configuration mode.

The **id** sets the short MA name.

Step 5 **maximum-meps** *number*

Example:

```
RP/0/(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/(config-cfm-dmn-svc)# log continuity-check errors
```

(Optional) Enables logging of certain types of events.

Step 7 **end** or **commit**

Example:

```
RP/0/(config-cfm-dmn-svc)# commit
```

Saves configuration changes.

Configuring CFM MEPs

When you configure CFM MEPs, consider these guidelines:

- Up to 1000 local MEPs are supported per card.
- CFM maintenance points can be created on All physical Ethernet interfaces (except for the RP Management interfaces).
- CFM maintenance points can be created on both Layer 2 interfaces.

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **interface** { **TenGigE | HundredGigE**} *interface-path-id*

Example:

```
RP/0/(config)# interface TenGigE 0/1/0/1
```

Type of Ethernet interface on which you want to create a MEP. Enter **TenGigE** or **HundredGigE** 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.

For more information about the syntax for the router, use the question mark (**?**) online help function.

Step 3 **interface** { **TenGigE | HundredGigE | Bundle-Ether**} *interface-path-id***.***subinterface*

Example:

```
RP/0/(config)# interface TenGigE 0/1/0/1
```

Type of Ethernet interface on which you want to create a MEP. Enter **TenGigE, HundredGigE** or **Bundle-Ether** and the physical interface or virtual interface followed by the subinterface path ID.

Naming notation is *interface-path-id***.***subinterface*. The period in front of the subinterface value is required as part of the notation.

For more information about the syntax for the router, use the question mark (**?**) online help function.

Step 4 **interface** {**FastEthernet | TenGigE | HundredGigE**} *interface-path-id*

Example:

```
RP/0/(config)# interface TenGigE 0/1/0/1
```

Type of Ethernet interface on which you want to create a MEP. Enter **FastEthernet, TenGigE** or **HundredGigE** 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.

For more information about the syntax for the router, use the question mark (**?**) online help function.

Step 5 **ethernet cfm**

Example:

```
RP/0/(config-if)# ethernet cfm
```

Enters interface Ethernet CFM configuration mode.

Step 6 **mep domain** *domain-name* **service** *service-name* **mep-id** *id-number*

Example:

```
RP/0/(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 7 **cos** *cos*

Example:

```
RP/0/(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.

Step 8 **end** or **commit**

Example:

```
RP/0/(config-if-cfm-mep)# commit
```

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.

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **ethernet cfm**

Example:

```
RP/0/(config)# ethernet cfm
```

Enters Ethernet CFM global configuration mode.

Step 3 **domain** *domain-name* **level** *level-value* **id null**

Example:

```
RP/0/(config-cfm)# domain Domain_One level 1 id null
```

Specifies the domain and domain level.

Step 4 **service** *name* **xconnect group** *xconnect-group-name* **p2p** *xconnect-name*

Example:

```
RP/0/(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2
```

Specifies the service and cross-connect group and name.

Step 5 **ais transmission** [**interval** {**1s**|**1m**}][**cos** *cos*]

Example:

```
RP/0/(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/(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/(config-sla-prof-stat-cfg)# commit
```

Saves configuration changes.

Configuring AIS on a CFM Interface

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

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **interface TenGigE** *interface-path-id*

Example:

```
RP/0/(config)# interface TenGigE 0/1/0/2
```

Enters interface configuration mode.

Step 3 **ethernet cfm**

Example:

```
RP/0/(config-if)# ethernet cfm
```

Enters Ethernet CFM interface configuration mode.

Step 4 **ais transmission up interval 1m cos** *cos*

Example:

```
RP/0/(config-if-cfm)# ais transmission up interval 1m cos 7
```

Configures Alarm Indication Signal (AIS) transmission on a Connectivity Fault Management (CFM) interface.

Step 5 **end** or **commit**

Example:

```
RP/0/(config-if-cfm)# commit
```

Saves configuration changes.

Configuring EFD for a CFM Service

To configure EFD for a CFM service, complete the following steps.

Restrictions

EFD is not supported on up MEPs. It can only be configured on down MEPs, within a particular service.

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **ethernet cfm**

Example:

```
RP/0/(config)# ethernet cfm
```

Enters CFM configuration mode.

Step 3 **domain** *domain-name* **level** *level-value* **id null**

Example:

```
RP/0/(config-cfm)# domain Domain_One level 1 id null
```

Specifies or creates the CFM domain and enters CFM domain configuration mode.

Step 4 **service** *service-name* **down-meps**

Example:

```
RP/0/(config-cfm-dmn)# service S1 down-meps
```

Specifies or creates the CFM service for down MEPS and enters CFM domain service configuration mode.

Step 5 **efd**

Example:

```
RP/0/(config-cfm-dmn-svc)# efd
```

Enables EFD on all down MEPs in the down MEPS service.

Step 6 **log efd**

Example:

```
RP/0/(config-cfm-dmn-svc)# log efd
```

(Optional) Enables logging of EFD state changes on an interface.

Step 7 **end** or **commit**

Example:

```
RP/0/(config-cfm-dmn-svc)# commit
```

Saves configuration changes.

Verifying the EFD Configuration

This example shows how to display all interfaces that are shut down because of Ethernet Fault Detection (EFD):

```
RP/0/# show efd interfaces

Server VLAN MA
==============
Interface         Clients
--------------------------
TenGigE0/0/0/0.0     CFM
```

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.

Troubleshooting Tips

To troubleshoot problems within the CFM network, perform the following steps:

Procedure

Step 1 To verify connectivity to a problematic MEP, use the **ping ethernet cfm** command as shown in the following example:

```
RP/0/# ping ethernet cfm domain D1 service S1 mep-id 16 source interface
TenGigE  0/0/0/0

Type escape sequence to abort.
Sending 5 CFM Loopbacks, timeout is 2 seconds -
Domain foo (level 2), Service foo
Source: MEP ID 1, interface TenGigE0/0/0/0
Target: 0001.0002.0003 (MEP ID 16):
  Running (5s) ...
Success rate is 60.0 percent (3/5), round-trip min/avg/max = 1251/1349/1402 ms
Out-of-sequence: 0.0 percent (0/3)
Bad data: 0.0 percent (0/3)
Received packet rate: 1.4 pps
```

Step 2 If the results of the **ping ethernet cfm** command show a problem with connectivity to the peer MEP, use the **traceroute ethernet cfm** command to help further isolate the location of the problem as shown in the following example:

```
RP/0/# traceroute ethernet cfm domain D1 service S1 mep-id 16 source
interface TenGigE 0/0/0/0

Traceroutes in domain D1 (level 4), service S1
Source: MEP-ID 1, interface TenGigE0/0/0/0
================================================================================
Traceroute at 2009-05-18 12:09:10 to 0001.0203.0402,
TTL 64, Trans ID 2:

Hop Hostname/Last            Ingress MAC/name       Egress MAC/Name        Relay
--- ---------------------- ---------------------- ---------------------- -----
  1 ios                    0001.0203.0400 [Down]                         FDB
      0000-0001.0203.0400    TenGigE0/0/0/0
  2 abc                                           0001.0203.0401 [Ok]    FDB
      ios                                         Not present
  3 bcd                    0001.0203.0402 [Ok]                           Hit
      abc                  TenGigE0/0
Replies dropped: 0
```

If the target was a MEP, verify that the last hop shows “Hit” in the Relay field to confirm connectivity to the peer MEP.

If the Relay field contains “MPDB” for any of the hops, then the target MAC address was not found in the bridge MAC learning table at that hop, and the result is relying on CCM learning. This result can occur under normal conditions, but it can also indicate a problem. If you used the **ping ethernet cfm** command before using the **traceroute ethernet cfm** command, then the MAC address should have been learned. If “MPDB” is appearing in that case, then this indicates a problem at that point in the network.

Configuring Ethernet LMI

To configure Ethernet LMI, complete the following tasks:

Prerequisites for Configuring E-LMI

Before you configure E-LMI on the Cisco NCS 4000 Series Router, be sure that you complete the following requirements:

- Identify the local and remote UNIs in your network where you want to run E-LMI, and define a naming convention for them.
- Enable E-LMI on the corresponding CE interface link on a device that supports E-LMI CE operation, such as the Cisco Catalyst 3750 Metro Series Switches.

Restrictions for Configuring E-LMI

When configuring E-LMI, consider the following restrictions:

- E-LMI is not supported on subinterfaces or bundle interfaces. E-LMI is configurable on Ethernet physical interfaces only.

Configuring UNI Names on the Physical Interface

It is recommended that you configure UNI names on the physical interface links to both the local and remote UNIs to aid in management for the E-LMI protocol. To configure UNI names, complete the following tasks on the physical interface links to both the local and remote UNIs:

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **interface [TenGigE | HundredGigE]** *interface-path-id*

Example:

```
RP/0/(config)# interface TenGigE 0/0/0/0
```

Enters interface configuration mode for the physical interface.

Step 3 **ethernet uni id** *name*

Example:

```
RP/0/(config-if)# ethernet uni id PE1-CustA-Slot0-Port0
```

Specifies a name (up to 64 characters) for the Ethernet UNI interface link.

Step 4 **end** or **commit**

Example:

```
RP/0/(config-if)# commit
```

Saves configuration changes.

Enabling E-LMI on the Physical Interface

It supports the E-LMI protocol only on physical Ethernet interfaces. To enable E-LMI, complete the following tasks on the physical Ethernet interface link to the local UNI:

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **interface** [**TenGigE** | **HundredGigE**] *interface-path-id*

Example:

```
RP/0/(config)# interface TenGigE 0/0/0/0
```

Enters interface configuration mode for the physical interface.

Step 3 **ethernet lmi**

Example:

```
RP/0/(config-if)# ethernet lmi
```

Enables Ethernet Local Management Interface operation on an interface and enters interface Ethernet LMI configuration mode.

Step 4 **end** or **commit**

Example:

```
RP/0/(config-if-lmi)# commit
```

Saves configuration changes.

Configuring the Status Counter

The MEF N393 Status Counter value is used to determine E-LMI operational status by tracking receipt of consecutive good packets or successive expiration of the PVT on packets. The default counter is four, which means that while the E-LMI protocol is in Down state, four good packets must be received consecutively to change the protocol state to Up, or while the E-LMI protocol is in Up state, four consecutive PVT expirations must occur before the state of the E-LMI protocol is changed to Down on the interface.

To modify the status counter default value, complete the following tasks:

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **interface** [**TenGigE** | **HundredGigE**] *interface-path-id*

Example:

```
RP/0/(config)# interface TenGigE 0/0/0/0
```

Enters interface configuration mode for the physical interface.

Step 3 **ethernet lmi**

Example:

```
RP/0/(config-if)# ethernet lmi
```

Enables Ethernet Local Managment Interface operation on an interface and enters interface Ethernet LMI configuration mode.

Step 4 **status-counter** *threshold*

Example:

```
RP/0/(config-if-lmi)# status-counter 5
```

Sets the MEF N393 Status Counter value that is used to determine E-LMI operational status by tracking receipt of consecutive good and bad packets from a peer. The default is 4.

Step 5 **end** or **commit**

Example:

```
RP/0/(config-if-lmi)# commit
```

Saves configuration changes.

Configuring the Polling Verification Timer

The MEF T392 Polling Verification Timer (PVT) specifies the allowable time between transmission of a STATUS message and receipt of a STATUS ENQUIRY from the UNI-C before recording an error. The default value is 15 seconds.

To modify the default value or disable the PVT altogether, complete the following tasks:

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **interface** [**TenGigE** | **HundredGigE**] *interface-path-id*

Example:

```
RP/0/# interface TenGigE 0/0/0/0
```

Enters interface configuration mode for the physical interface.

Step 3 **ethernet lmi**

Example:

```
RP/0/(config-if)# ethernet lmi
```

Enables Ethernet Local Managment Interface operation on an interface and enters interface Ethernet LMI configuration mode.

Step 4 **polling-verification-timer** {*interval* | **disable**}

Example:

```
RP/0/(config-if-lmi)# polling-verification-timer 30
```

Sets or disables the MEF T392 Polling Verification Timer for E-LMI operation, which specifies the allowable time (in seconds) between transmission of a STATUS message and receipt of a STATUS ENQUIRY from the UNI-C before recording an error. The default is 15.

Step 5 **end** or **commit**

Example:

```
RP/0/(config-if-lmi)# commit
```

Saves configuration changes.

Disabling Syslog Messages for E-LMI Errors or Events

The E-LMI protocol tracks certain errors and events whose counts can be displayed using the **show ethernet lmi interfaces** command.

To disable syslog messages for E-LMI errors or events, complete the following tasks:

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **interface** [**TenGigE** | **HundredGigE**] *interface-path-id*

Example:

```
RP/0/(config)# interface TenGigE 0/0/0/0
```

Enters interface configuration mode for the physical interface.

Step 3 **ethernet lmi**

Example:

```
RP/0/(config-if)# ethernet lmi
```

Enables Ethernet Local Managment Interface operation on an interface and enters interface Ethernet LMI configuration mode.

Step 4 **log** {**errors** | **events**} **disable**

Example:

```
RP/0/(config-if-lmi)# log events disable
```

Turns off syslog messages for E-LMI errors or events.

Step 5 **end** or **commit**

Example:

```
RP/0/(config-if-lmi)# commit
```

Saves configuration changes.

Disabling Use of the Cisco-Proprietary Remote UNI Details Information Element

To provide additional information within the E-LMI protocol, the Cisco IOS XR software implements a Cisco-proprietary information element called Remote UNI Details to send information to the CE about remote UNI names and states. This information element implements what is currently an unused identifier from the E-LMI MEF 16 specification.

To disable use of the Remote UNI Details information element, complete the following tasks:

Procedure

Step 1 **configure**

Example:

```
RP/0/# configure
```

Enters global configuration mode.

Step 2 **interface** [**TenGigE** | **HundredGigE**] *interface-path-id*

Example:

```
RP/0/(config)# interface TenGigE 0/0/0/0
```

Enters interface configuration mode for the physical interface.

Step 3 **ethernet lmi**

Example:

```
RP/0/(config-if)# ethernet lmi
```

Enables Ethernet Local Managment Interface operation on an interface and enters interface Ethernet LMI configuration mode.

Step 4 **extension remote-uni disable**

Example:

```
RP/0/(config-if-lmi)# extension remote-uni disable
```

Disables transmission of the Cisco-proprietary Remote UNI Details information element in E-LMI STATUS messages.

Step 5 **end** or **commit**

Example:

```
RP/0/(config-if-lmi)# commit
```

Saves configuration changes.

Verifying the Ethernet LMI Configuration

Use the **show ethernet lmi interfaces detail** command to display the values for the Ethernet LMI configuration for a particular interface, or for all interfaces. The following example shows sample output for the command:

```
RP/0/# show ethernet lmi interfaces detail
Interface: TenGigE0/0/0/0
  Ether LMI Link Status: Up
  UNI Id: PE1-CustA-Slot0-Port0
  Line Protocol State: Up
```

```
MTU: 1514 (1 PDU reqd. for full report)
CE-VLAN/EVC Map Type: Bundling (1 EVC)
Configuration: Status counter 4, Polling Verification Timer 15 seconds
Last Data Instance Sent: 0
Last Sequence Numbers: Sent 0, Received 0

Reliability Errors:
  Status Enq Timeouts                  0 Invalid Sequence Number             0
  Invalid Report Type                  0

Protocol Errors:
  Malformed PDUs                       0 Invalid Procotol Version            0
  Invalid Message Type                 0 Out of Sequence IE                  0
  Duplicated IE                        0 Mandatory IE Missing                0
  Invalid Mandatory IE                 0 Invalid non-Mandatory IE            0
  Unrecognized IE                      0 Unexpected IE                       0

Full Status Enq Received     never       Full Status Sent         never
PDU Received                 never       PDU Sent                 never
LMI Link Status Changed   00:00:03 ago   Last Protocol Error      never
Counters cleared             never

Sub-interface: TenGigE0/0/0/0.0
  VLANs: 1-20
  EVC Status: Active
  EVC Type: Point-to-Point
  OAM Protocol: CFM
    CFM Domain: Global (level 5)
    CFM Service: CustomerA
  Remote UNI Count: Configured = 1, Active = 1

  Remote UNI Id                                                    Status
  -------------                                                    ------
  PE1-CustA-Slot0-Port1                                            Up
```

Configuration Examples for Ethernet OAM

This section provides the following configuration examples:

Configuration Examples for EOAM Interfaces

This section provides the following configuration examples:

Configuring an Ethernet OAM Profile Globally: Example

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

```
configure terminal
 ethernet oam profile Profile_1
  link-monitor
   symbol-period window 60000
   symbol-period threshold low 10000000 high 60000000
   frame window 60
   frame threshold low 10000000 high 60000000
   frame-period window 60000
   frame-period threshold low 100 high 12000000
   frame-seconds window 900000
   frame-seconds threshold 3 threshold 900
```

```
 exit
mib-retrieval
connection timeout 30
require-remote mode active
require-remote link-monitoring
require-remote mib-retrieval
action dying-gasp error-disable-interface
action critical-event error-disable-interface
action discovery-timeout error-disable-interface
action session-down error-disable-interface
action capabilities-conflict error-disable-interface
action wiring-conflict error-disable-interface
action remote-loopback error-disable-interface
commit
```

Configuring Ethernet OAM Features on an Individual Interface: Example

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

```
configure terminal
 interface TenGigE 0/1/0/0
  ethernet oam
   link-monitor
    symbol-period window 60000
    symbol-period threshold low 10000000 high 60000000
    frame window 60
    frame threshold low 10000000 high 60000000
    frame-period window 60000
    frame-period threshold low 100 high 12000000
    frame-seconds window 900000
    frame-seconds threshold 3 threshold 900
    exit
   mib-retrieval
   connection timeout 30
   require-remote mode active
   require-remote link-monitoring
   require-remote mib-retrieval
   action link-fault  error-disable-interface
   action dying-gasp error-disable-interface
   action critical-event error-disable-interface
   action discovery-timeout error-disable-interface
   action session-down error-disable-interface
   action capabilities-conflict error-disable-interface
   action wiring-conflict error-disable-interface
   action remote-loopback 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 terminal
 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
  action remote-loopback disable
  action uni-directional link-fault error-disable-interface
  commit

configure terminal
 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
    action remote-loopback log
    action uni-directional link-fault log
    uni-directional link-fault detection
    commit
```

Configuring a Remote Loopback on an Ethernet OAM Peer: Example

This example shows how to configure a remote loopback on an Ethernet OAM peer:

```
RP/0/# configure terminal
RP/0/(config)# interface TenGigE 0/1/5/6
RP/0/(config-if)# ethernet oam
RP/0/(config-if-eoam)# profile Profile_1
RP/0/(config-if-eoam)# remote-loopback
RP/0/(config-if-eoam)# commit
```

This example shows how to start a remote loopback on a configured Ethernet OAM interface:

```
RP/0/# ethernet oam loopback enable TenGigE 0/1/5/6
```

Clearing Ethernet OAM Statistics on an Interface: Example

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

```
RP/0/# clear ethernet oam statistics interface TenGigE 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 terminal
  ethernet oam profile Profile_1
  snmp-server traps ethernet oam events
```

Configuration Examples for Ethernet CFM

This section includes the following examples:

Ethernet CFM Domain Configuration: Example

This example shows how to configure a basic domain for Ethernet CFM:

```
configure
ethernet cfm
traceroute cache hold-time 1 size 3000
domain Domain_One level 1 id null
commit
```

Ethernet CFM Service Configuration: Example

The following examples show how to create a service for an Ethernet CFM Service:

```
RP/0/(config-cfm-dmn)# service Bridge_Service down-meps number 10
RP/0/(config-cfm-dmn)# commit
```

```
RP/0/(config-cfm-dmn)# service S1 xconnect group XG1 p2p X1
RP/0/(config-cfm-dmn)# commit
```

Continuity Check for an Ethernet CFM Service Configuration: Example

This example shows how to configure continuity-check options for an Ethernet CFM service:

```
continuity-check archive hold-time 100
continuity-check loss auto-traceroute
continuity-check interval 100ms loss-threshold 10
commit
```

MIP Creation for an Ethernet CFM Service Configuration: Example

This example shows how to enable MIP auto-creation for an Ethernet CFM service:

```
RP/0/(config-cfm-dmn-svc)# mip auto-create all
RP/0/(config-cfm-dmn-svc)# commit
```

Cross-check for an Ethernet CFM Service Configuration: Example

This example shows how to configure cross-check for MEPs in an Ethernet CFM service:

```
mep crosscheck
 mep-id 10
 mep-id 20
 commit
```

Other Ethernet CFM Service Parameter Configuration: Example

This example shows how to configure other Ethernet CFM service options:

```
 maximum-meps 4000
 log continuity-check errors
 commit
 exit
exit
exit
```

MEP Configuration: Example

This example shows how to configure a MEP for Ethernet CFM on an interface:

```
RP/0/# configure
RP/0/(config)# interface TenGigE 0/1/0/1
RP/0/(config-if)# ethernet cfm
RP/0/(config-if-cfm)# mep domain Dm1 service Sv1 mep-id 1
RP/0/(config-if-cfm-mep)# commit
```

Ethernet CFM Show Command: Examples

These examples show how to verify the configuration of Ethernet Connectivity Fault Management (CFM):

Example 1

This example shows how to display all the maintenance points that have been created on an interface:

```
RP/0/# show ethernet cfm local maintenance-points

Domain/Level         Service              Interface                Type      ID          MAC
-------------------- -------------------- -----------------       ----   --------    -------
fig/5                bay                  TenGigE0/10/0/12.23456   Dn MEP    2
44:55:66
fig/5                bay                  TenGigE0/0/1/0.1         MIP
55:66:77
fred/3               barney               TenGigE0/1/0/0.1         Up MEP    5
66:77:88!
```

Example 2

This example shows how to display all the CFM configuration errors on all domains:

```
RP/0/# show ethernet cfm configuration-errors

Domain fig (level 5), Service bay
* MIP creation configured using bridge-domain blort, but bridge-domain blort does not exist.

 * An Up MEP is configured for this domain on interface TenGigE0/1/2/3.234 and an Up MEP
is also configured for domain blort, which is at the same level (5).
 * A MEP is configured on interface TenGigE0/3/2/1.1 for this domain/service, which has CC
 interval 100ms, but the lowest interval supported on that interface is 1s
```

Example 3

This example shows how to display operational state for local maintenance end points (MEPs):

```
RP/0/# show ethernet cfm local meps

 A - AIS received                I - Wrong interval
 R - Remote Defect received      V - Wrong Level
 L - Loop (our MAC received)     T - Timed out (archived)
 C - Config (our ID received)    M - Missing (cross-check)
 X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
 P - Peer port down

Domain foo (level 6), Service bar
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
```

```
----- ------------------------ --- -------- -- ------- ---
  100 TenGigE1/1/0/1.234 (Up)  Up     0/0   N  A       L7

Domain fred (level 5), Service barney
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
    2 TenGigE0/1/0/0.234 (Up)  Up     3/2   Y  RPC     L6
Domain foo (level 6), Service bar
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
  100 TenGigE1/1/0/1.234 (Up)  Up     0/0   N  A

Domain fred (level 5), Service barney
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
    2 TenGigE0/1/0/0.234 (Up)  Up     3/2   Y  RPC
```

Example 4

This example shows how to display operational state of other maintenance end points (MEPs) detected by a local MEP:

```
RP/0/# show ethernet cfm peer meps

Flags:
 > - Ok                          I - Wrong interval
 R - Remote Defect received      V - Wrong level
 L - Loop (our MAC received)     T - Timed out
 C - Config (our ID received)    M - Missing (cross-check)
 X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)

Domain fred (level 7), Service barney
Up MEP on TenGigE0/1/0/0.234, MEP-ID 2
================================================================================
St    ID MAC address    Port    Up/Downtime   CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ----------- --------- ------ ----- -----
 >     1 0011.2233.4455 Up      00:00:01         1234      0     0     0
R>     4 4455.6677.8899 Up      1d 03:04         3456      0   234     0
L      2 1122.3344.5566 Up      3w 1d 6h         3254      0     0  3254
C      2 7788.9900.1122 Test    00:13            2345      6    20  2345
X      3 2233.4455.6677 Up      00:23              30      0     0    30
I      3 3344.5566.7788 Down    00:34           12345      0   300  1234
V      3 8899.0011.2233 Blocked 00:35              45      0     0    45
 T     5 5566.7788.9900         00:56              20      0     0     0
M      6                                           0      0     0     0
U>     7 6677.8899.0011 Up      00:02             456      0     0     0

Domain fred (level 7), Service fig
Down MEP on TenGigE0/10/0/12.123, MEP-ID 3
================================================================================
St    ID MAC address    Port    Up/Downtime  CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ----------- -------- ------ ----- -----
 >     1 9900.1122.3344 Up      03:45           4321      0     0     0
```

Example 5

This example shows how to display operational state of other maintenance end points (MEPs) detected by a local MEP with details:

```
RP/0/#  show ethernet cfm peer meps detail
Domain dom3 (level 5), Service ser3
```

```
Down MEP on TenGigE0/0/0/0 MEP-ID 1
==============================================================================
Peer MEP-ID 10, MAC 0001.0203.0403
   CFM state: Wrong level, for 00:01:34
   Port state: Up
   CCM defects detected:    V - Wrong Level
   CCMs received: 5
     Out-of-sequence:             0
     Remote Defect received:      5
     Wrong Level:                 0
     Cross-connect (wrong MAID):  0
     Wrong Interval:              5
     Loop (our MAC received):     0
     Config (our ID received):    0
Last CCM received 00:00:06 ago:
     Level: 4, Version: 0, Interval: 1min
     Sequence number: 5, MEP-ID: 10
     MAID: String: dom3, String: ser3
     Port status: Up, Interface status: Up


Domain dom4 (level 2), Service ser4
Down MEP on TenGigE0/0/0/0 MEP-ID 1
==============================================================================
Peer MEP-ID 20, MAC 0001.0203.0402
   CFM state: Ok, for 00:00:04
   Port state: Up
   CCMs received: 7
     Out-of-sequence:             1
     Remote Defect received:      0
     Wrong Level:                 0
     Cross-connect (wrong MAID):  0
     Wrong Interval:              0
     Loop (our MAC received):     0
  Config (our ID received):    0
Last CCM received 00:00:04 ago:
     Level: 2, Version: 0, Interval: 10s
     Sequence number: 1, MEP-ID: 20
     MAID: String: dom4, String: ser4
     Chassis ID: Local: ios; Management address: 'Not specified'
     Port status: Up, Interface status: Up

Peer MEP-ID 21, MAC 0001.0203.0403
   CFM state: Ok, for 00:00:05
   Port state: Up
   CCMs received: 6
     Out-of-sequence:             0
     Remote Defect received:      0
     Wrong Level:                 0
     Cross-connect (wrong MAID):  0
     Wrong Interval:              0
     Loop (our MAC received):     0
     Config (our ID received):    0
Last CCM received 00:00:05 ago:
     Level: 2, Version: 0, Interval: 10s
     Sequence number: 1, MEP-ID: 21
     MAID: String: dom4, String: ser4
     Port status: Up, Interface status: Up


Domain dom5 (level 2), Service ser5
Up MEP on Standby Bundle-Ether 1 MEP-ID 1
==============================================================================
Peer MEP-ID 600, MAC 0001.0203.0401
```

```
   CFM state: Ok (Standby), for 00:00:08, RDI received
   Port state: Down
   CCM defects detected:    Defects below ignored on local standby MEP
                            I - Wrong Interval
                            R - Remote Defect received
   CCMs received: 5
     Out-of-sequence:             0
     Remote Defect received:    5
  Wrong Level:                 0
     Cross-connect W(wrong MAID): 0
     Wrong Interval:              5
     Loop (our MAC received):     0
     Config (our ID received):    0
   Last CCM received 00:00:08 ago:
     Level: 2, Version: 0, Interval: 10s
     Sequence number: 1, MEP-ID: 600
     MAID: DNS-like: dom5, String: ser5
     Chassis ID: Local: ios; Management address: 'Not specified'
     Port status: Up, Interface status: Down

Peer MEP-ID 601, MAC 0001.0203.0402
   CFM state: Timed Out (Standby), for 00:15:14, RDI received
   Port state: Down
   CCM defects detected:    Defects below ignored on local standby MEP
                            I - Wrong Interval
                            R - Remote Defect received
                            T - Timed Out
                            P - Peer port down
   CCMs received: 2
     Out-of-sequence:             0
     Remote Defect received:      2
     Wrong Level:                 0
     Cross-connect (wrong MAID):  0
     Wrong Interval:              2
     Loop (our MAC received):     0
     Config (our ID received):    0
   Last CCM received 00:15:49 ago:
     Level: 2, Version: 0, Interval: 10s
     Sequence number: 1, MEP-ID: 600
     MAID: DNS-like: dom5, String: ser5
     Chassis ID: Local: ios; Management address: 'Not specified'
     Port status: Up, Interface status: Down
```

AIS for CFM Configuration: Examples

Example 1

This example shows how to configure Alarm Indication Signal (AIS) transmission for a CFM domain service:

```
RP/0/# configure
RP/0/(config)# ethernet cfm
RP/0/(config-cfm)# domain D1 level 1
RP/0/(config-cfm-dmn)# service Cross_Connect_1 xconnect group XG1 p2p
RP/0/(config-cfm-dmn-svc)# ais transmission interval 1m cos 7
```

Example 2

This example shows how to configure AIS logging for a Connectivity Fault Management (CFM) domain service to indicate when AIS or LCK packets are received:

```
RP/0/# configure
RP/0/(config)# ethernet cfm
RP/0/(config-cfm)# domain D1 level 1
RP/0/(config-cfm-dmn)# service Cross_Connect_1 xconnect group XG1 p2p
RP/0/(config-cfm-dmn-svc)# log ais
```

This example shows how to configure AIS transmission on a CFM interface.

```
RP/0/# configure
RP/0/(config)# interface TenGigE 0/1/0/2
RP/0/(config-if)# ethernet cfm
RP/0/(config-if-cfm)# ais transmission up interval 1m cos 7
```

AIS for CFM Show Commands: Examples

This section includes the following examples:

show ethernet cfm interfaces ais Command: Example

This example shows how to display the information published in the Interface AIS table:

```
RP/0/# show ethernet cfm interfaces ais

Defects (from at least one peer MEP):
 A - AIS received                I - Wrong interval
 R - Remote Defect received      V - Wrong Level
 L - Loop (our MAC received)     T - Timed out (archived)
 C - Config (our ID received)    M - Missing (cross-check)
 X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
 P - Peer port down              D - Local port down

                          Trigger                  Transmission
                    AIS   ---------    Via    --------------------------
Interface (State)   Dir   L Defects  Levels   L Int Last started  Packets
------------------------ ---  - -------  -------  - --- ------------ --------
TenGi0/1/0/0.234 (Up)    Dn   5 RPC      6        7 1s  01:32:56 ago     5576
TenGi0/1/0/0.567 (Up)    Up   0 M        2,3      5 1s  00:16:23 ago      983
TenGi0/1/0/1.1 (Dn)      Up     D                7 60s 01:02:44 ago     3764
TenGi0/1/0/2 (Up)        Dn   0 RX       1!
```

show ethernet cfm local meps Command: Examples

Example 1: Default

The following example shows how to display statistics for local maintenance end points (MEPs):

```
RP/0/# show ethernet cfm local meps

 A - AIS received                I - Wrong interval
 R - Remote Defect received      V - Wrong Level
 L - Loop (our MAC received)     T - Timed out (archived)
 C - Config (our ID received)    M - Missing (cross-check)
 X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
 P - Peer port down

Domain foo (level 6), Service bar
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
```

```
  100 TenGigE1/1/0/1.234 (Up)  Up     0/0   N  A       7

Domain fred (level 5), Service barney
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
    2 TenGigE0/1/0/0.234 (Up)  Up     3/2   Y  RPC     6
```

Example 2: Domain Service

The following example shows how to display statistics for MEPs in a domain service:

```
RP/0/# show ethernet cfm local meps domain foo service bar detail

Domain foo (level 6), Service bar
Up MEP on TenGigE0/1/0/0.234, MEP-ID 100
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)

  CCM generation enabled:  No
  AIS generation enabled:  Yes (level: 7, interval: 1s)
  Sending AIS:             Yes (started 01:32:56 ago)
  Receiving AIS:           Yes (from lower MEP, started 01:32:56 ago)

Domain fred (level 5), Service barney
Up MEP on TenGigE0/1/0/0.234, MEP-ID 2
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 3 up, 2 with errors, 0 timed out (archived)
  Cross-check defects: 0 missing, 0 unexpected

  CCM generation enabled:  Yes (Remote Defect detected: Yes)
  CCM defects detected:    R - Remote Defect received
                           P - Peer port down
                           C - Config (our ID received)
  AIS generation enabled:  Yes (level: 6, interval: 1s)
  Sending AIS:             Yes (to higher MEP, started 01:32:56 ago)
  Receiving AIS:           No
```

Example 3: Verbose

The following example shows how to display verbose statistics for MEPs in a domain service:

Note The Discarded CCMs field is not displayed when the number is zero (0). It is unusual for the count of discarded CCMs to be any thing other than zero, since CCMs are only discarded when the limit on the number of peer MEPs is reached.

```
RP/0/# show ethernet cfm local meps domain foo service bar verbose

Domain foo (level 6), Service bar
Up MEP on TenGigE0/1/0/0.234, MEP-ID 100
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)

  CCM generation enabled:  No
  AIS generation enabled:  Yes (level: 7, interval: 1s)
  Sending AIS:             Yes (started 01:32:56 ago)
```

```
  Receiving AIS:           Yes (from lower MEP, started 01:32:56 ago)

  Packet        Sent      Received
  ------  ----------  ----------------------------------------------------
  CCM             20            20  (out of seq: 0)
  AIS           5576            0

Domain fred (level 5), Service barney
Up MEP on TenGigE0/1/0/0.234, MEP-ID 2
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 3 up, 2 with errors, 0 timed out (archived)
  Cross-check defects: 0 missing, 0 unexpected

  CCM generation enabled:  Yes (Remote Defect detected: Yes)
  CCM defects detected:    R - Remote Defect received
                           P - Peer port down
                           C - Config (our ID received)
  AIS generation enabled:  Yes (level: 6, interval: 1s)
  Sending AIS:             Yes (to higher MEP, started 01:32:56 ago)
  Receiving AIS:           No

  Packet        Sent      Received
  ------  ----------  ------------------------------------------------------------
  CCM          12345         67890  (out of seq: 6, discarded: 10)
  LBM              5             0
  LBR              0             5  (out of seq: 0, with bad data: 0)
  AIS              0         46910
  LCK              -             0
```

Example 4: Detail

The following example shows how to display detailed statistics for MEPs in a domain service:

```
RP/0/# show ethernet cfm local meps detail

Domain foo (level 6), Service bar
Up MEP on TenGigE0/1/0/0.234, MEP-ID 100
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)

  CCM generation enabled:  No
  AIS generation enabled:  Yes (level: 7, interval: 1s)
  Sending AIS:             Yes (started 01:32:56 ago)
  Receiving AIS:           Yes (from lower MEP, started 01:32:56 ago)

Domain fred (level 5), Service barney
Up MEP on TenGigE0/1/0/0.234, MEP-ID 2
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 3 up, 2 with errors, 0 timed out (archived)
  Cross-check defects: 0 missing, 0 unexpected

  CCM generation enabled:  Yes (Remote Defect detected: Yes)
  CCM defects detected:    R - Remote Defect received
                           P - Peer port down
                           C - Config (our ID received)
  AIS generation enabled:  Yes (level: 6, interval: 1s)
  Sending AIS:             Yes (to higher MEP, started 01:32:56 ago)
  Receiving AIS:           No
```

CFM - Sample Configuration Workflow

Complete these configurations on the provider edge routers to enable Connectivity Fault Management (CFM).

Topology

```
----(Te0/3/0/11)NCS4K-PE1(Hu0/5/0/0)-----(Hu0/5/0/0)NCS4K-PE2(Te0/5/0/9)----
```

where:

- TenGigE0/3/0/11 and TenGigE0/5/0/9 are the access or customer interfaces
- The HundredGigE0/5/0/0 interfaces are the core interfaces.
- PE1 and PE2 are the two L2VPN provider edge (PE) routers. The two PEs are typically connected at two different sites with an MPLS core between them. The attachment circuits (ACs)connected at each L2VPN PE are linked by a pseudowire (PW) over the MPLS network.

Task 1: Bring up the controllers in lan phy or packet termination mode.

Sample Configuration on PE1	Sample Configuration on PE2
!	!
controller Optics0/3/0/11	controller Optics0/5/0/9
port-mode Ethernet framing packet rate 10GE	port-mode Ethernet framing packet rate 10GE
no shut	no shut
! controller Optics0/5/0/0	! controller Optics0/5/0/0
port-mode Ethernet framing packet rate 100GE	port-mode Ethernet framing packet rate 100GE
no shut	no shut
!	!

Task 2: Bring up the access and core interfaces.

Sample Configuration on PE1	Sample Configuration on PE2

Access interface:
 	
`interface TenGigE0/3/0/11`
`!`
`interface TenGigE0/3/0/11.1 l2transport`
`encapsulation dot1q 1`
`no shut`
`!`
`interface TenGigE0/3/0/11.2 l2transport`
`encapsulation dot1q 2`
`no shut`
`!`
 	
Core interface:
 	
`interface HundredGigE0/5/0/0`
`ipv4 address 1.76.1.1 255.255.255.0`
`!`
`!`
	Access interface:
	
	`interface TenGigE0/5/0/9`
	`!`
	`interface TenGigE0/5/0/9.1 l2transport`
	`encapsulation dot1q 1`
	`no shut`
	`!`
	`interface TenGigE0/5/0/9.2 l2transport`
	`encapsulation dot1q 2`
	`no shut`
	`!`
	
	Core interface:
	
	`interface HundredGigE0/5/0/0`
	`ipv4 address 1.76.1.2 255.255.255.0`
	`!`
	`!`

Details: Two access interfaces are brought up so that two pseudowires can be created.	

Task 3: Define loopback address.

Sample Configuration on PE1	Sample Configuration on PE2

`!`
`interface Loopback0`
`ipv4 address 1.1.1.1 255.255.255.255`
`!`
	`!`
	`interface Loopback0`
	`ipv4 address 3.3.3.3 255.255.255.255`
	`!`

Task 4: Configure the routing process using OSPF or ISIS on the core interface.

Sample Configuration on PE1	Sample Configuration on PE2

router ospf 100
router-id 1.1.1.1
nsf
nsr
 area 0
 mpls traffic-eng
 interface Loopback0
  !
  interface HundredGigE0/5/0/0
  !
 !
 mpls traffic-eng router-id Loopback0
 !
	router ospf 100
	router-id 3.3.3.3
	nsf
	nsr
	 area 0
	 mpls traffic-eng
	 interface Loopback0
	  !
	  interface HundredGigE0/5/0/0
	  !
	 !
	 mpls traffic-eng router-id Loopback0
	 !

Details: The sample configuration uses OSPF.	

Task 5: Configure MPLS traffic engineering on the core interface.

Sample Configuration on PE1	Sample Configuration on PE2
mpls traffic-eng interface HundredGigE0/5/0/0 ! fault-oam !	mpls traffic-eng interface HundredGigE0/5/0/0 ! fault-oam !

Task 6: Configure RSVP on the core interface.

Sample Configuration on PE1	Sample Configuration on PE2
rsvp interface HundredGigE0/5/0/0 bandwidth 100 ! !	rsvp interface HundredGigE0/5/0/0 bandwidth 100 ! !

Task 7: Configure MPLS OAM for MPLS pseudowires to work on the core interfaces.

Sample Configuration on PE1	Sample Configuration on PE2
!	!
mpls oam	mpls oam
!	!

Task 8: Configure the tunnel interface. It can be a MPLS-TE or Flex-LSP tunnel.

Sample Configuration on PE1	Sample Configuration on PE2

Sample Configuration on PE1	Sample Configuration on PE2

MPLS TE tunnel:
 	
`interface tunnel-te1`
`ipv4 unnumbered Loopback0`
`signalled-bandwidth 1`
`destination 3.3.3.3`
`path-selection`
`metric te`
`bandwidth 50000`
`!`
` path-option 1 dynamic`
`!`
Flex-LSP tunnel with BFD:
 	
`interface tunnel-te2`
`ipv4 unnumbered Loopback0`
`bfd`
` encap-mode gal`
` multiplier 3`
` fast-detect`
` minimum-interval 100`
` !`
` signalled-bandwidth 1`
` destination 3.3.3.3`
` bidirectional`
` association id 86`
`source-address 192.0.0.0`
` association type co-routed`
`  fault-oam`
`  !`
` !`
`path-selection`
`  metric te`
	MPLS TE tunnel:
	
	`interface tunnel-te1`
	`ipv4 unnumbered Loopback0`
	`signalled-bandwidth 1`
	`destination 1.1.1.1`
	`path-selection`
	`metric te`
	`bandwidth 50000`
	`!`
	` path-option 1 dynamic`
	`!`
	Flex-LSP tunnel with BFD:
	
	`interface tunnel-te2`
	`ipv4 unnumbered Loopback0`
	`bfd`
	` encap-mode gal`
	` multiplier 3`
	` fast-detect`
	` minimum-interval 100`
	` !`
	` signalled-bandwidth 1`
	` destination 1.1.1.1`
	` bidirectional`
	` association id 86`
	`source-address 192.0.0.0`
	` association type co-routed`
	`  fault-oam`
	`  !`
	` !`
	`path-selection`
	`  metric te`

Sample Configuration on PE1	Sample Configuration on PE2
bandwidth 50000 ! path-option 1 dynamic !	bandwidth 50000 ! path-option 1 dynamic !

Task 9: Setup interfaces running LDP:

Sample Configuration on PE1	Sample Configuration on PE2

mpls ldp
 	
nsr
 	
log
 	
  neighbor
 	
   nsr
 	
  graceful-restart
 	
  !
 	
graceful-restart reconnect-timeout 169
 	
graceful-restart forwarding-
state-holdtime 180
 	
discovery
 	
targeted-hello holdtime 180
 	
targeted-hello interval 20
 	
!
 	
router-id 1.1.1.1
 	
session protection
 	
address-family ipv4
 	
discovery targeted-hello accept
 	
 !
!
	mpls ldp
	
	nsr
	
	log
	
	  neighbor
	
	   nsr
	
	  graceful-restart
	
	  !
	
	graceful-restart reconnect-timeout 169
	
	graceful-restart forwarding-
	state-holdtime 180
	
	discovery
	
	targeted-hello holdtime 180
	
	targeted-hello interval 20
	
	!
	
	router-id 3.3.3.3
	
	session protection
	
	address-family ipv4
	
	discovery targeted-hello accept
	
	 !
	!

Details: The two PEs establish a targeted MPLS LDP session between themselves so they can establish and control the status of the pseudowire. The targeted MPLS LDP session is established over MPLS-TE or Flex LSP.	

Task 10: Configure VPWS static and dynamic pseudowires.

Sample Configuration on PE1	Sample Configuration on PE2

Pseudowire 1 (vpws-pw-1) uses MPLS-TE tunnel (tunnel-te 1):
 	
`l2vpn`
`pw-class vpws-pw-1`
`encapsulation mpls`
`protocol ldp`
`ipv4 source 1.1.1.1`
`preferred-path interface tunnel-te 1`
` !`
	Pseudowire 1 (vpws-pw-1) uses MPLS-TE tunnel (tunnel-te 1):
	
	`l2vpn`
	`pw-class vpws-pw-1`
	`encapsulation mpls`
	`protocol ldp`
	`ipv4 source 3.3.3.3`
	`preferred-path interface tunnel-te 1`
	` !`

Pseudowire 2 (vpws-pw-2) uses Flex-LSP tunnel (tunnel-te 2):
 	
`!`
` pw-class vpws-pw-2`
` encapsulation mpls`
` protocol ldp`
` ipv4 source 1.1.1.1`
` preferred-path interface tunnel-te 2`
`!`
	Pseudowire 2 (vpws-pw-2) uses Flex-LSP tunnel (tunnel-te 2):
	
	`!`
	` pw-class vpws-pw-2`
	` encapsulation mpls`
	` protocol ldp`
	` ipv4 source 3.3.3.3`
	` preferred-path interface tunnel-te 2`
	`!`

Configure pseudowire 1 (vpws-pw-1) as dynamic:
 	
`!`
` xconnect group vpws`
` p2p pw1`
` interface TenGigE0/3/0/11.1`
` neighbor ipv4 3.3.3.3 pw-id 1`
` bandwidth 1000`
` pw-class vpws-pw-1`
` !`
	Configure pseudowire 1 (vpws-pw-1) as dynamic:
	
	`!`
	` xconnect group vpws`
	` p2p pw1`
	` interface TenGigE0/5/0/9.1`
	` neighbor ipv4 1.1.1.1 pw-id 1`
	` bandwidth 1000`
	` pw-class vpws-pw-1`
	` !`

Sample Configuration on PE1	Sample Configuration on PE2
Configure pseudowire 2(vpws-pw-2) as static: ! p2p pw2 interface TenGigE0/3/0/11.2 neighbor ipv4 3.3.3.3 pw-id 2 mpls static label local 100 remote 200 bandwidth 1000 pw-class vpws-pw-2 ! !	Configure pseudowire 2(vpws-pw-2) as static: ! p2p pw2 interface TenGigE0/5/0/9.2 neighbor ipv4 1.1.1.1 pw-id 2 mpls static label local 100 remote 200 bandwidth 1000 pw-class vpws-pw-2 ! !

Task 11: Configure Connectivity Fault Management (CFM).

Sample Configuration on PE1	Sample Configuration on PE2
	l2vpn xconnect group xc1 p2p pw2 interface TenGigE0/5/0/9 interface HundredGigE0/5/0/0 ! !
domain MD2 level 2 id null service up_mep_customer_1 xconnect group xc1 p2p p1 id number 1 continuity-check interval 100ms mep crosscheck mep-id 4001 mac-address 7ef2.fe69.312b !	domain MD2 level 2 id null service up_mep_customer_1 xconnect group xc1 p2p p1 id number 1 continuity-check interval 100ms mep crosscheck mep-id 1 mac-address 78ba.f99b.b9ea !

Sample Configuration on PE1	Sample Configuration on PE2
ais transmission interval 1s cos 0 log ais log continuity-check errors log crosscheck errors log continuity-check mep changes ! !	ais transmission interval 1s cos 0 log ais log continuity-check errors log crosscheck errors log continuity-check mep changes ! !
	interface TenGigE0/5/0/9 l2transport encapsulation dot1q 1 ethernet cfm mep domain MD2 service up_mep_customer_1 mep-id 4001 ! ! !
interface TenGigE0/3/0/11 l2transport encapsulation dot1q 1 ethernet cfm mep domain MD2 service up_mep_customer_1 mep-id 1 ! ! !	interface HundredGigE0/5/0/0 l2transport encapsulation dot1q 1 ethernet cfm mep domain MD1 service down_mep_customer_10001 mep-id 4001 ! ! !

EFD Configuration: Examples

This example shows how to enable EFD:

```
RP/0/# configure
RP/0/(config)# ethernet cfm
RP/0/(config-cfm)# domain D1 level 1 id null
```

```
RP/0/(config-cfm-dmn)# service S1 down-meps id number 1
RP/0/(config-cfm-dmn-svc)# efd
```

This example shows how to enable EFD logging:

```
RP/0/# configure
RP/0/(config)# ethernet cfm
RP/0/(config-cfm)# domain D1 level 1 id null
RP/0/(config-cfm-dmn)# service S1 down-meps id number 1
RP/0/(config-cfm-dmn-svc)# log efd
```

Displaying EFD Information: Examples

The following examples show how to display information about EFD:

show efd interfaces Command: Example

This example shows how to display all interfaces that are shut down in response to an EFD action:

```
RP/0/# show efd interfaces

Server VLAN MA
==============
Interface         Clients
------------------------
TenGigE0/0/0/0.0     CFM
```

show ethernet cfm local meps detail Command: Example

Use the **show ethernet cfm local meps detail** command to display MEP-related EFD status information. The following example shows that EFD is triggered for MEP-ID 100:

```
RP/0/# show ethernet cfm local meps detail

Domain foo (level 6), Service bar
Up MEP on TenGigE0/1/0/0.234, MEP-ID 100
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)
  Cross-check errors: 2 missing, 0 unexpected

  CCM generation enabled:  No
  AIS generation enabled:  Yes (level: 7, interval: 1s)
  Sending AIS:             Yes (started 01:32:56 ago)
  Receiving AIS:           Yes (from lower MEP, started 01:32:56 ago)
  EFD triggered:           Yes

Domain fred (level 5), Service barney
Up MEP on TenGigE0/1/0/0.234, MEP-ID 2
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 3 up, 0 with errors, 0 timed out (archived)
  Cross-check errors: 0 missing, 0 unexpected

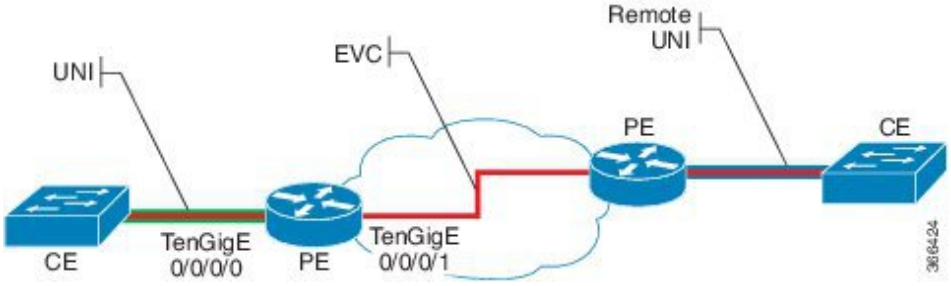
  CCM generation enabled:  Yes (Remote Defect detected: No)
  AIS generation enabled:  Yes (level: 6, interval: 1s)
  Sending AIS:             No
  Receiving AIS:           No
  EFD triggered:           No
```

Note You can also verify that EFD has been triggered on an interface using the **show interfaces** and **show interfaces brief** commands. When an EFD trigger has occurred, these commands will show the interface status as *up* and the line protocol state as *down*.

Configuration Example for Ethernet LMI

Figure below shows a basic E-LMI network environment with a local UNI defined as the PE using Ten-Gigabit Ethernet interface 0/0/0/0, and connectivity to a remote UNI over Ten-Gigabit Ethernet interface 0/0/0/1.

Figure 15: Basic E-LMI UNI and Remote UNI Diagram



The following configuration provides a basic E-LMI configuration for the environment shown in figure above, as the PE device on the local UNI with physical Ten-Gigabit Ethernet interfaces 0/0/0/0 and 0/0/0/1:

```
RP/0/# configure
!
! Configure the Local UNI EFPs
!
RP/0/(config)# interface TenGigE0/0/0/0.0 l2transport
RP/0/(config-subif)#encapsulation dot1q 1-20
RP/0/(config-subif)# exit
RP/0/(config)# interface TenGigE0/0/0/1.1 l2transport
RP/0/(config-subif)# #encapsulation dot1q 1-20
RP/0/(config-subif)# exit
!
! Create the EVC
!
RP/0/(config)# l2vpn
RP/0/(config-l2vpn)# bridge group BG1
RP/0/(config-l2vpn-bg)# bridge-domain BD1
RP/0/(config-l2vpn-bg-bd)# interface TenGigE0/0/0/0.0
RP/0/(config-l2vpn-bg-bd)# interface TenGigE0/0/0/1.1
RP/0/(config-l2vpn-bg-bd)# exit
RP/0/(config-l2vpn-bg)# exit
RP/0/(config-l2vpn)# exit
!
! Configure Ethernet CFM
!
RP/0/(config)# ethernet cfm
RP/0/(config-cfm)# domain GLOBAL level 5
RP/0/(config-cfm-dmn)# service CustomerA bridge group BG1 bridge-domain BD1
RP/0/(config-cfm-dmn-svc)# continuity-check interval 100ms
RP/0/(config-cfm-dmn-svc)# mep crosscheck mep-id 22
RP/0/(config-cfm-dmn-svc)# mep crosscheck mep-id 11
RP/0/(config-cfm-dmn-svc)# exit
```

```
RP/0/(config-cfm-dmn)# exit
RP/0/(config-cfm)# exit
!
! Configure EFPs as CFM MEPs
!
RP/0/(config)# interface TenGigE0/0/0/0.0 l2transport
RP/0/(config-subif)# ethernet cfm
RP/0/(config-if-cfm)# mep domain GLOBAL service CustomerA mep-id 22
RP/0/(config-if-cfm)# exit
RP/0/(config-subif)# exit
!
! Configure the Local UNI Name
!
RP/0/(config)# interface TenGigE 0/0/0/0
RP/0/(config-if)# ethernet uni id PE1-CustA-Slot0-Port0
RP/0/(config-if)# exit
!
! Enable E-LMI on the Local UNI Physical Interface
!
RP/0/(config)# interface TenGigE 0/0/0/0
RP/0/(config-if)# ethernet lmi
RP/0/(config-if)# exit
RP/0/(config)# commit
```