Configure Ethernet OAM

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

- Prerequisites for Configuring Ethernet OAM, on page 1
- Restrictions for Configuring Ethernet OAM, on page 1
- Information About Configuring Ethernet OAM, on page 2
- How to Configure Ethernet OAM, on page 21
- Configuration Examples for Ethernet OAM, on page 50

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

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.
- CFM does not support Y.1731 Performance Measurement i.e Delay Measurement(DMM), Loss Measurement(LMM) and Synthetic Measurement(SLM).
- 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 cannot 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).

### 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.
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.
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.

Figure 2: Different CFM Maintenance Domains Across a Network

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

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

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

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

Note

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

Maintenance Points

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

A maintenance point is always associated with a particular CFM service, and therefore with a particular maintenance domain at a particular level. Maintenance points generally only process CFM frames at the same level as their associated maintenance domain. Frames at a higher maintenance level are always forwarded transparently, while frames at a lower maintenance level are normally dropped. This helps enforce the maintenance domain hierarchy described in the Maintenance Domains, 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.
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.

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.

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 4: 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
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.

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 5: 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 6: 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.
In IEEE 802.1ag, the CCM Learning Database is referred to as the MIP CCM Database. However, it applies to both MIPs and MEPs.

Note
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.
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.
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 9.
- 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.
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 9: CFM Error Detection and EFD Trigger

Figure 10: 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).

## 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 11: 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/RP0:hostname# configure terminal

Enters global configuration mode.

Step 2 ethernet oam profile profile-name

Example:

RP/0/RP0:hostname(config)# ethernet oam profile Profile_1

Creates a new Ethernet Operations, Administration and Maintenance (OAM) profile and enters Ethernet OAM configuration mode.

Step 3 link-monitor

Example:

RP/0/RP0:hostname(config-eoam)# link-monitor

Enters the Ethernet OAM link monitor configuration mode.

Step 4 symbol-period window window

Example:

RP/0/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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/RP0:hostname(config-eoam-lm)# exit
Exits back to Ethernet OAM mode.

Step 13  mib-retrieval

Example:

RP/0/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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}
Configuring an Ethernet OAM Profile

**Example:**

RP/0/RP0:hostname(config-eoam)# mode passive

Configures the Ethernet OAM mode. The default is active.

**Step 17**

```
require-remote mode {active|passive}
```

**Example:**

RP/0/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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/RP0:hostname(config-eoam)# action capabilities-conflict efd

Specifies the action that is taken on an interface when a capabilities-conflict event occurs. The default action is to create a syslog entry.

**Note**

- If you change the default, the log keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.

**Step 21**

```
action critical-event {disable | error-disable-interface}
```

**Example:**

RP/0/RP0:hostname(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}
```
Configure Ethernet OAM

Example:

RP/0/RP0:hostname(config-eoam)# action discovery-timeout efd

Specifies the action that is taken on an interface when a connection timeout occurs. The default action is to create a syslog entry.

Note: • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.

Step 23: action dying-gasp \{disable | error-disable-interface\}

Example:

RP/0/RP0:hostname(config-eoam)# action dying-gasp error-disable-interface

Specifies the action that is taken on an interface when a dying-gasp notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry.

Note: • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.

Step 24: action high-threshold \{error-disable-interface | log\}

Example:

RP/0/RP0:hostname(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/RP0:hostname(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/RP0:hostname(config-eoam)# action session-down efd

Configure Ethernet OAM Profile
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/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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/RP0:hostname(config-if)# commit
```
Saves the configuration changes to the running configuration file and remains within the configuration session.

**Step 32**

*end*

**Example:**

```
RP/0/RP0:hostname(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/RP0:hostname# configure terminal
```

Enters global configuration mode.

**Step 2**

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

**Example:**

```
RP/0/RP0:hostname(config)# interface TenGigE 0/1/0/0
```

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

**Note**

- The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.

**Step 3**

*ethernet oam*

**Example:**

```
RP/0/RP0:hostname(config-if)# ethernet oam
```

Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.

**Step 4**

*profile profile-name*

**Example:**

```
RP/0/RP0:hostname(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:**
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.

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

**Procedure**

**Step 1** configure
Example:

RP/0/RP0:hostname# configure terminal
Enters global configuration mode.

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

RP/0/RP0:hostname(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/RP0:hostname(config-if)# ethernet oam

Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.

Step 4  interface-Ethernet-OAM-command
Example:

RP/0/RP0:hostname(config-if-eoam)# action capabilities-conflict error-disable-interface

Configures a setting for an Ethernet OAM configuration command and overrides the setting for the profile configuration, where interface-Ethernet-OAM-command is one of the supported commands on the platform in interface Ethernet OAM configuration mode.

Step 5  commit
Example:

RP/0/RP0:hostname(config-if)# commit

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

Step 6  end
Example:

RP/0/RP0:hostname(config-if)# end

Ends the configuration session and exits to the EXEC mode.

---

Verifying the Ethernet OAM Configuration

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

RP/0/RP0:hostname# 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
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/RP0:hostname# configure
```

Enters global configuration mode.

**Step 2** ethernet cfm

Example:

```
RP/0/RP0:hostname# ethernet cfm
```

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

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

Example:

```
RP/0/RP0:hostname(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.
Configuring a CFM Maintenance Domain

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

Procedure

Step 1  configure
Example:

RP/0/RP0:hostname# configure
Enters global configuration mode.

Step 2  ethernet cfm
Example:
RP/0/RP0:hostname(config)# ethernet cfm
Enters Ethernet Connectivity Fault Management (CFM) configuration mode.

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

**Example:**

RP/0/RP0:hostname(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/RP0:hostname(config-cfm)# traceroute cache hold-time 1 size 3000

(Optional) Sets the maximum limit of traceroute cache entries or the maximum time limit to hold the traceroute cache entries. The default is 100 minutes and 100 entries.

### Step 5
end  or  commit

**Example:**

RP/0/RP0:hostname(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/RP0:hostname# configure

Enters global configuration mode.

### Step 2
ethernet cfm

**Example:**

RP/0/RP0:hostname# ethernet cfm
Step 3  \textbf{domain} \textit{domain-name} \textbf{level} \textit{level-value} \textbf{id} \textit{null}

\textbf{Example:}

RP/0/RP0:hostname(config-cfm)# domain Domain_One level 1 id null

This command creates and names a container for all domain configurations at a specified maintenance level, and enters CFM domain configuration mode.

The \textit{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  \textbf{service} \textit{service-name} \[ \textbf{down-meps} \mid \textbf{xconnect} \] \textbf{id} \[ \textbf{icc-based} \textit{icc-string} \mid \textbf{number} \textit{number} \]

\textbf{Example:}

RP/0/RP0:hostname(config-cfm-dmn)# service Bridge_Service down-meps number 10

This command configures and associates a service with the domain and enters CFM domain service configuration mode.

The \textit{id} sets the short MA name.

Step 5  \textbf{end} or \textbf{commit}

\textbf{Example:}

RP/0/RP0:hostname(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.

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  \textbf{configure}

\textbf{Example:}

RP/0/RP0:hostname# configure
Enters global configuration mode.

**Step 2**
**ethernet cfm**

Example:

```
RP/0/RP0:hostname(config)# ethernet cfm
```

Enters Ethernet Connectivity Fault Management (CFM) configuration mode.

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

Example:

```
RP/0/RP0:hostname(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/RP0:hostname(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/RP0:hostname(config-cfm-dmn-svc)# continuity-check interval 100m loss-threshold 10
```

(Optional) Enables Continuity Check and specifies the time interval at which CCMs are transmitted or to set the threshold limit for when a MEP is declared down.

**Step 6**
**continuity-check archive hold-time minutes**

Example:

```
RP/0/RP0:hostname(config-cfm-dmn-svc)# continuity-check archive hold-time 100
```

(Optional) Configures how long information about peer MEPs is stored after they have timed out.

**Step 7**
**continuity-check loss auto-traceroute**

Example:

```
RP/0/RP0:hostname(config-cfm-dmn-svc)# continuity-check loss auto-traceroute
```

(Optional) Configures automatic triggering of a traceroute when a MEP is declared down.

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

Example:
RP/0/RP0:hostname(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/RP0:hostname# configure
Enters global configuration mode.

Step 2 ethernet cfm
Example:
RP/0/RP0:hostname# ethernet cfm
Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.

Step 3 domain domain-name level level-value id null
Example:
RP/0/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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/RP0:hostname# configure  
Enter global configuration mode.

Step 2  
**ethernet cfm**  

Example:

RP/0/RP0:hostname(config)# ethernet cfm  
Enter the Ethernet Connectivity Fault Management (CFM) configuration mode.

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

Example:

RP/0/RP0:hostname(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**  

Example:

RP/0/RP0:hostname(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:
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/RP0:hostname# configure
Enters global configuration mode.

Step 2 interface { TenGigE | HundredGigE } interface-path-id
Example:

RP/0/RP0:hostname(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.
Step 3

```
interface { TenGigE | HundredGigE | Bundle-Ether} interface-path-id.subinterface
```

**Example:**
```
RP/0/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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:**
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/RP0:hostname(config-if-cfm-mep)# configure
```

Enters global configuration mode.

**Step 2** ethernet cfm

*Example:*

```
RP/0/RP0:hostname(config)# ethernet cfm
```

Enters Ethernet CFM global configuration mode.

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

*Example:*

```
RP/0/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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
Configuring AIS on a CFM Interface

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

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>RP/0/RP0:hostname# configure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>interface TenGigE interface-path-id</td>
<td>RP/0/RP0:hostname(config)# interface TenGigE 0/1/0/2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>3</td>
<td>ethernet cfm</td>
<td>RP/0/RP0:hostname(config-if)# ethernet cfm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enters Ethernet CFM interface configuration mode.</td>
</tr>
<tr>
<td>4</td>
<td>ais transmission up interval 1m cos</td>
<td>RP/0/RP0:hostname(config-if-cfm)# ais transmission up interval 1m cos 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configures Alarm Indication Signal (AIS) transmission on a CFM interface.</td>
</tr>
<tr>
<td>5</td>
<td>end or commit</td>
<td>RP/0/RP0:hostname(config-cfm-dmn-svc)# log ais</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configures AIS logging for a Connectivity Fault Management (CFM) domain service to indicate when AIS or LCK packets are received.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>RP/0/RP0:hostname(config-sla-prof-stat-cfg)# commit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RP0:hostname(config-cfm-dmn-svc)# log ais
```

Configuring AIS on a CFM Interface

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

**Procedure**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>RP/0/RP0:hostname# configure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>interface TenGigE interface-path-id</td>
<td>RP/0/RP0:hostname(config)# interface TenGigE 0/1/0/2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enters interface configuration mode.</td>
</tr>
<tr>
<td>3</td>
<td>ethernet cfm</td>
<td>RP/0/RP0:hostname(config-if)# ethernet cfm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enters Ethernet CFM interface configuration mode.</td>
</tr>
<tr>
<td>4</td>
<td>ais transmission up interval 1m cos</td>
<td>RP/0/RP0:hostname(config-if-cfm)# ais transmission up interval 1m cos 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configures Alarm Indication Signal (AIS) transmission on a CFM interface.</td>
</tr>
<tr>
<td>5</td>
<td>end or commit</td>
<td>RP/0/RP0:hostname(config-cfm-dmn-svc)# log ais</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Configures AIS logging for a Connectivity Fault Management (CFM) domain service to indicate when AIS or LCK packets are received.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
<td>RP/0/RP0:hostname(config-sla-prof-stat-cfg)# commit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>
RP/0/RP0:hostname(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/RP0:hostname# configure
Enters global configuration mode.

**Step 2**

*ethernet cfm*

**Example:**

RP/0/RP0:hostname(config)# ethernet cfm
Enters CFM configuration mode.

**Step 3**

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

**Example:**

RP/0/RP0:hostname(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/RP0:hostname(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/RP0:hostname(config-cfm-dmn-svc)# efd
Enables EFD on all down MEPs in the down MEPS service.

**Step 6**

*log efd*
Example:

RP/0/RP0:hostname(config-cfm-dmn-svc)# log efd

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

Step 7

end or commit

Example:

RP/0/RP0:hostname(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/RP0:hostname# 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:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show ethernet cfm configuration-errors [domain domain-name] [interface interface-path-id]</td>
<td>Displays information about errors that are preventing configured CFM operations from becoming active, as well as any warnings that have occurred.</td>
</tr>
<tr>
<td>show ethernet cfm local maintenance-points domain name [service name]</td>
<td>Displays a list of local maintenance points.</td>
</tr>
<tr>
<td>[interface interface-path-id] [mep</td>
<td>mip]</td>
</tr>
</tbody>
</table>

---

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/RP0:hostname# ping ethernet cfm domain D1 service S1 mep-id 16 source interface TenGigE 0/0/0/0
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/RP0:hostname# 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 0000-0001.0203.0400 [Down] FDB
  2   abc 0001.0203.0401 [Ok] FDB
  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/RP0:hostname# configure
```

Enters global configuration mode.

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

Example:

```
RP/0/RP0:hostname(config)# interface TenGigE 0/0/0/0
```

Enters interface configuration mode for the physical interface.

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

Example:

```
RP/0/RP0:hostname(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/RP0:hostname(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/RP0:hostname# configure

Enters global configuration mode.

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

**Example:**
RP/0/RP0:hostname(config)# interface TenGigE 0/0/0/0

Enters interface configuration mode for the physical interface.

**Step 3**
ethernet lmi

**Example:**
RP/0/RP0:hostname(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/RP0:hostname(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/RP0:hostname# configure
Enters global configuration mode.

Step 2 interface [TenGigE | HundredGigE] interface-path-id
Example:

RP/0/RP0:hostname(config)# interface TenGigE 0/0/0/0
Enters interface configuration mode for the physical interface.

Step 3 ethernet lmi
Example:

RP/0/RP0:hostname(config-if)# ethernet lmi
Enables Ethernet Local Management Interface operation on an interface and enters interface Ethernet LMI configuration mode.

Step 4 status-counter threshold
Example:

RP/0/RP0:hostname(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/RP0:hostname(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/RP0:hostname# configure
Enters global configuration mode.

**Step 2**

`interface [TenGigE | HundredGigE] interface-path-id`

**Example:**

```
RP/0/RP0:hostname# interface TenGigE 0/0/0/0
```

Enters interface configuration mode for the physical interface.

**Step 3**

`ethernet lmi`

**Example:**

```
RP/0/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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/RP0:hostname# configure
```

Enters global configuration mode.

**Step 2**

`interface [TenGigE | HundredGigE] interface-path-id`

Example:
Example:

```
RP/0/RP0:hostname(config)# interface TenGigE 0/0/0/0
```
Enters interface configuration mode for the physical interface.

**Step 3**

ethernet lmi

Example:

```
RP/0/RP0:hostname(config-if)# ethernet lmi
```
Enables Ethernet Local Management Interface operation on an interface and enters interface Ethernet LMI configuration mode.

**Step 4**

log {errors | events} disable

Example:

```
RP/0/RP0:hostname(config-if-lmi)# log events disable
```
Turns off syslog messages for E-LMI errors or events.

**Step 5**

end or commit

Example:

```
RP/0/RP0:hostname(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/RP0:hostname# configure
```
Enters global configuration mode.

**Step 2**

interface [TenGigE | HundredGigE] interface-path-id

Example:

```
RP/0/RP0:hostname(config)# interface TenGigE 0/0/0/0
```
Enters interface configuration mode for the physical interface.

**Step 3**  
**ethernet lmi**  
**Example:**

```
RP/0/RP0:hostname(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/RP0:hostname(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/RP0:hostname(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/RP0:hostname# 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 Report Type        0

  Protocol Errors:
    Malformed PDUs             0
    Invalid Message Type       0
    Duplicated IE              0
    Invalid Mandatory IE       0
    Unrecognized IE            0

    Full Status Enq Received   never
    PDU Received               never
    LMI Link Status Changed    00:00:03 ago
```

Configure Ethernet OAM
Counts 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

<table>
<thead>
<tr>
<th>Remote UNI Id</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE1-CustA-Slot0-Port1</td>
<td>Up</td>
</tr>
</tbody>
</table>

**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
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 1000000 high 60000000
    frame window 60
    frame threshold low 1000000 high 60000000
    frame-period window 60000
    frame-period threshold low 100 high 12000000
    frame-seconds window 900000
    frame-seconds threshold low 10 high 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
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/RP0:hostname# configure terminal
RP/0/RP0:hostname(config)# interface TenGigE 0/1/5/6
RP/0/RP0:hostname(config-if)# ethernet oam
RP/0/RP0:hostname(config-if-eoam)# profile Profile_1
RP/0/RP0:hostname(config-if-eoam)# remote-loopback
RP/0/RP0:hostname(config-if-eoam)# commit

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

RP/0/RP0:hostname# 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/RP0:hostname# 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
eternet 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/RP0:hostname(config-cfm-dmn)# service Bridge_Service down-meps number 10
RP/0/RP0:hostname(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:

```plaintext
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:

```plaintext
RP/0/RP0:hostname(config-cfm-dmn-svc)# mip auto-create all
RP/0/RP0:hostname(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:

```plaintext
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:

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

MEP Configuration: Example

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

```plaintext
RP/0/RP0:hostname# configure
RP/0/RP0:hostname(config)# interface TenGigE 0/1/0/1
RP/0/RP0:hostname(config-if)# ethernet cfm
RP/0/RP0:hostname(config-if-cfm)# mep domain Dm1 service Sv1 mep-id 1
RP/0/RP0:hostname(config-if-cfm-mep)# commit
```

Ethernet CFM Show Command: Examples

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

```plaintext
```
Example 1

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

RP/0/RP0:hostname# show ethernet cfm local maintenance-points

<table>
<thead>
<tr>
<th>Domain/Level</th>
<th>Service</th>
<th>Interface</th>
<th>Type</th>
<th>ID</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>fig/5 44:55:66</td>
<td>bay</td>
<td>TenGigE0/10/0.12.23456</td>
<td>Dn MEP</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>fig/5 55:66:77</td>
<td>bay</td>
<td>TenGigE0/0/1/0.1</td>
<td>MIP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fred/3 66:77:88</td>
<td>barney</td>
<td>TenGigE0/1/0/0.1</td>
<td>Up MEP</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

Example 2

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

RP/0/RP0:hostname# 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/RP0:hostname# 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/RP0:hostname# 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
```

<table>
<thead>
<tr>
<th>St</th>
<th>ID</th>
<th>MAC address</th>
<th>Port</th>
<th>Up/Downtime</th>
<th>CcmRcvd</th>
<th>SeqErr</th>
<th>RDI Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>1</td>
<td>0011.2233.4455</td>
<td>Up</td>
<td>00:00:01</td>
<td>1234</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>R&gt;</td>
<td>4</td>
<td>4455.6677.8899</td>
<td>Up</td>
<td>0d 03:04</td>
<td>3456</td>
<td>0</td>
<td>234</td>
</tr>
<tr>
<td>L</td>
<td>2</td>
<td>1122.3344.5566</td>
<td>Up</td>
<td>3w 01:06</td>
<td>3254</td>
<td>0</td>
<td>0 3254</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>7788.9900.1122</td>
<td>Test</td>
<td>00:13</td>
<td>2345</td>
<td>6</td>
<td>20 2345</td>
</tr>
<tr>
<td>X</td>
<td>3</td>
<td>2233.4455.6677</td>
<td>Up</td>
<td>00:23</td>
<td>30</td>
<td>0</td>
<td>0 30</td>
</tr>
<tr>
<td>I</td>
<td>3</td>
<td>3344.5566.7788</td>
<td>Down</td>
<td>00:34</td>
<td>12345</td>
<td>0</td>
<td>300 1234</td>
</tr>
<tr>
<td>V</td>
<td>3</td>
<td>8899.0011.2233</td>
<td>Blocked</td>
<td>00:35</td>
<td>45</td>
<td>0</td>
<td>0 45</td>
</tr>
<tr>
<td>T</td>
<td>5</td>
<td>5566.7788.9900</td>
<td>Up</td>
<td>00:56</td>
<td>20</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>M</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>U&gt;</td>
<td>7</td>
<td>6677.8899.0011</td>
<td>Up</td>
<td>00:02</td>
<td>456</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Domain fred (level 7), Service fig
Down MEP on TenGigE0/10/0/0.234, MEP-ID 3

```
```

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/RP0:hostname# show ethernet cfm peer meps detail
```

```
Domain dom3 (level 5), Service ser3
Down MEP on TenGigE0/10/0/0.234, MEP-ID 3
```

```
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
```

```
```

Configure Ethernet OAM
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
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 00:00:08 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/RP0:hostname# configure
RP/0/RP0:hostname(config)# ethernet cfm
RP/0/RP0:hostname(config-cfm)# domain D1 level 1
RP/0/RP0:hostname(config-cfm-dmn)# service Cross_Connect_1 xconnect group XG1 p2p
RP/0/RP0:hostname(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/RP0:hostname# configure
RP/0/RP0:hostname(config)# ethernet cfm
RP/0/RP0:hostname(config-cfm)# domain D1 level 1
RP/0/RP0:hostname(config-cfm-dmn)# service Cross_Connect_1 xconnect group XG1 p2p
RP/0/RP0:hostname(config-cfm-dmn-svc)# log ais
```

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

```
RP/0/RP0:hostname# configure
RP/0/RP0:hostname(config)# interface TenGigE 0/1/0/2
RP/0/RP0:hostname(config-if)# ethernet cfm
RP/0/RP0:hostname(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/RP0:hostname# 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
```

<table>
<thead>
<tr>
<th>Interface (State)</th>
<th>Dir</th>
<th>L Defects</th>
<th>Levels</th>
<th>Last started</th>
<th>Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>TenGi0/1/0/0.234 (Up)</td>
<td>Dn</td>
<td>5 RPC</td>
<td>6</td>
<td>01:32:56 ago</td>
<td>5576</td>
</tr>
<tr>
<td>TenGi0/1/0/0.567 (Up)</td>
<td>Up</td>
<td>0 M</td>
<td>2,3</td>
<td>00:16:23 ago</td>
<td>983</td>
</tr>
<tr>
<td>TenGi0/1/0/1.1 (Dn)</td>
<td>Up</td>
<td>D</td>
<td>7</td>
<td>01:02:44 ago</td>
<td>3764</td>
</tr>
<tr>
<td>TenGi0/1/0/2 (Up)</td>
<td>Dn</td>
<td>0 RX</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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/RP0:hostname# 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
```

Example 2: Domain Service

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

```
RP/0/RP0:hostname# 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)
```
Example 3: Verbose

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

```
RP/0/RP0:hostname# show ethernet cfm local meps domain foo service bar verbose
```
Sending AIS: Yes (to higher MEP, started 01:32:56 ago)
Receiving AIS: No

<table>
<thead>
<tr>
<th>Packet</th>
<th>Sent</th>
<th>Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCM</td>
<td>12345</td>
<td>67890</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(out of seq: 6, discarded: 10)</td>
</tr>
<tr>
<td>LBM</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>LBR</td>
<td>0</td>
<td>5 (out of seq: 0, with bad data: 0)</td>
</tr>
<tr>
<td>AIS</td>
<td>0</td>
<td>46910</td>
</tr>
<tr>
<td>LCK</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>

Example 4: Detail

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

```
RP/0/RP0:hostname# show ethernet cfm local meps detail
```

Domain foo (level 6), Service bar
Up MEP on TenGigE0/1/0/0.234, MEP-ID 100

<table>
<thead>
<tr>
<th>Interface state: Up</th>
<th>MAC address: 1122.3344.5566</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Interface state: Up</th>
<th>MAC address: 1122.3344.5566</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer MEPs: 3 up, 2 with errors, 0 timed out (archived)</td>
<td></td>
</tr>
<tr>
<td>Cross-check defects: 0 missing, 0 unexpected</td>
<td></td>
</tr>
</tbody>
</table>

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

```plaintext
----(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.
## Task 2: Bring up the access and core interfaces.

<table>
<thead>
<tr>
<th>Sample Configuration on PE1</th>
<th>Sample Configuration on PE2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access interface:</strong></td>
<td><strong>Access interface:</strong></td>
</tr>
<tr>
<td>interface TenGigE0/3/0/11</td>
<td>interface TenGigE0/5/0/9</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>interface TenGigE0/3/0/11.1</td>
<td>interface TenGigE0/5/0/9.1</td>
</tr>
<tr>
<td>l2transport</td>
<td>l2transport</td>
</tr>
<tr>
<td>encapsulation dot1q 1</td>
<td>encapsulation dot1q 1</td>
</tr>
<tr>
<td>no shut</td>
<td>no shut</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>interface TenGigE0/3/0/11.2</td>
<td>interface TenGigE0/5/0/9.2</td>
</tr>
<tr>
<td>l2transport</td>
<td>l2transport</td>
</tr>
<tr>
<td>encapsulation dot1q 2</td>
<td>encapsulation dot1q 2</td>
</tr>
<tr>
<td>no shut</td>
<td>no shut</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
</tbody>
</table>

**Core interface:**

interface HundredGigE0/5/0/0
ipv4 address 1.76.1.1 255.255.255.0

**Details:** Two access interfaces are brought up so that two pseudowires can be created.
### Task 3: Define loopback address.

<table>
<thead>
<tr>
<th>Sample Configuration on PE1</th>
<th>Sample Configuration on PE2</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>interface Loopback0</td>
<td>interface Loopback0</td>
</tr>
<tr>
<td>ipv4 address 1.1.1.1 255.255.255.255</td>
<td>ipv4 address 3.3.3.3 255.255.255.255</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Sample Configuration on PE1</th>
<th>Sample Configuration on PE2</th>
</tr>
</thead>
<tbody>
<tr>
<td>router ospf 100</td>
<td>router ospf 100</td>
</tr>
<tr>
<td>router-id 1.1.1.1</td>
<td>router-id 3.3.3.3</td>
</tr>
<tr>
<td>nsf</td>
<td>nsf</td>
</tr>
<tr>
<td>nsr</td>
<td>nsr</td>
</tr>
<tr>
<td>area 0</td>
<td>area 0</td>
</tr>
<tr>
<td>mpls traffic-eng</td>
<td>mpls traffic-eng</td>
</tr>
<tr>
<td>interface Loopback0</td>
<td>interface Loopback0</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>interface HundredGigE0/5/0/0</td>
<td>interface HundredGigE0/5/0/0</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>mpls traffic-eng router-id Loopback0</td>
<td>mpls traffic-eng router-id Loopback0</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
</tbody>
</table>

**Details:** The sample configuration uses OSPF.

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

<table>
<thead>
<tr>
<th>Sample Configuration on PE1</th>
<th>Sample Configuration on PE2</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpls traffic-eng</td>
<td>mpls traffic-eng</td>
</tr>
<tr>
<td>interface HundredGigE0/5/0/0</td>
<td>interface HundredGigE0/5/0/0</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>fault-oam</td>
<td>fault-oam</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
</tbody>
</table>

### Task 6: Configure RSVP on the core interface.
### Task 7: Configure MPLS OAM for MPLS pseudowires to work on the core interfaces.

<table>
<thead>
<tr>
<th>Sample Configuration on PE1</th>
<th>Sample Configuration on PE2</th>
</tr>
</thead>
<tbody>
<tr>
<td>rsvp</td>
<td>rsvp</td>
</tr>
<tr>
<td>interface HundredGigE0/5/0/0</td>
<td>interface HundredGigE0/5/0/0</td>
</tr>
<tr>
<td>bandwidth 100</td>
<td>bandwidth 100</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Sample Configuration on PE1</th>
<th>Sample Configuration on PE2</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>mpls oam</td>
<td>mpls oam</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>Sample Configuration on PE1</td>
<td>Sample Configuration on PE2</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------------------</td>
</tr>
</tbody>
</table>

**Sample Configuration Workflow**

Configure Ethernet OAM
<table>
<thead>
<tr>
<th>Sample Configuration on PE1</th>
<th>Sample Configuration on PE2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MPLS TE tunnel:</strong></td>
<td><strong>MPLS TE tunnel:</strong></td>
</tr>
<tr>
<td>interface tunnel-te1</td>
<td>interface tunnel-te1</td>
</tr>
<tr>
<td>ipv4 unnumbered Loopback0</td>
<td>ipv4 unnumbered Loopback0</td>
</tr>
<tr>
<td>signalled-bandwidth 1</td>
<td>signalled-bandwidth 1</td>
</tr>
<tr>
<td>destination 3.3.3.3</td>
<td>destination 1.1.1.1</td>
</tr>
<tr>
<td>path-selection</td>
<td>path-selection</td>
</tr>
<tr>
<td>metric te</td>
<td>metric te</td>
</tr>
<tr>
<td>bandwidth 50000</td>
<td>bandwidth 50000</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>path-option 1 dynamic</td>
<td>path-option 1 dynamic</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td><strong>Flex-LSP tunnel with BFD:</strong></td>
<td><strong>Flex-LSP tunnel with BFD:</strong></td>
</tr>
<tr>
<td>interface tunnel-te2</td>
<td>interface tunnel-te2</td>
</tr>
<tr>
<td>ipv4 unnumbered Loopback0</td>
<td>ipv4 unnumbered Loopback0</td>
</tr>
<tr>
<td>bfd</td>
<td>bfd</td>
</tr>
<tr>
<td>encap-mode gal</td>
<td>encap-mode gal</td>
</tr>
<tr>
<td>multiplier 3</td>
<td>multiplier 3</td>
</tr>
<tr>
<td>fast-detect</td>
<td>fast-detect</td>
</tr>
<tr>
<td>minimum-interval 100</td>
<td>minimum-interval 100</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>signalled-bandwidth 1</td>
<td>signalled-bandwidth 1</td>
</tr>
<tr>
<td>destination 3.3.3.3</td>
<td>destination 1.1.1.1</td>
</tr>
<tr>
<td>bidirectional</td>
<td>bidirectional</td>
</tr>
<tr>
<td>association id 86</td>
<td>association id 86</td>
</tr>
<tr>
<td>source-address 192.0.0.0</td>
<td>source-address 192.0.0.0</td>
</tr>
<tr>
<td>association type co-routed</td>
<td>association type co-routed</td>
</tr>
<tr>
<td>fault-oam</td>
<td>fault-oam</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>path-selection</td>
<td>path-selection</td>
</tr>
</tbody>
</table>
| metric te                   | metric te                   

Configure Ethernet OAM

CFM - Sample Configuration Workflow
Task 9: Setup interfaces running LDP:

<table>
<thead>
<tr>
<th>Sample Configuration on PE1</th>
<th>Sample Configuration on PE2</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>mpls ldp</code></td>
<td><code>mpls ldp</code></td>
</tr>
<tr>
<td><code>nsr</code></td>
<td><code>nsr</code></td>
</tr>
<tr>
<td><code>log</code></td>
<td><code>log</code></td>
</tr>
<tr>
<td><code>neighbor</code></td>
<td><code>neighbor</code></td>
</tr>
<tr>
<td><code>nsr</code></td>
<td><code>nsr</code></td>
</tr>
<tr>
<td><code>graceful-restart</code></td>
<td><code>graceful-restart</code></td>
</tr>
<tr>
<td><code>graceful-restart reconnect-timeout 169</code></td>
<td><code>graceful-restart reconnect-timeout 169</code></td>
</tr>
<tr>
<td><code>graceful-restart forwarding-state-holdtime 180</code></td>
<td><code>graceful-restart forwarding-state-holdtime 180</code></td>
</tr>
<tr>
<td><code>discovery</code></td>
<td><code>discovery</code></td>
</tr>
<tr>
<td><code>targeted-hello holdtime 180</code></td>
<td><code>targeted-hello holdtime 180</code></td>
</tr>
<tr>
<td><code>targeted-hello interval 20</code></td>
<td><code>targeted-hello interval 20</code></td>
</tr>
<tr>
<td><code>router-id 1.1.1.1</code></td>
<td><code>router-id 3.3.3.3</code></td>
</tr>
<tr>
<td><code>session protection</code></td>
<td><code>session protection</code></td>
</tr>
<tr>
<td><code>address-family ipv4</code></td>
<td><code>address-family ipv4</code></td>
</tr>
<tr>
<td><code>discovery targeted-hello accept</code></td>
<td><code>discovery targeted-hello accept</code></td>
</tr>
</tbody>
</table>

**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

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 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
```

Configure pseudowire 1 (vpws-pw-1) as dynamic:

```
! 
xconnect group vpws
p2p pw1
interface TenGigE0/5/0/9.1
neighbor ipv4 3.3.3.3 pw-id 1
bandwidth 1000
pw-class vpws-pw-1
```

---

### 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 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 3.3.3.3
preferred-path interface tunnel-te 2
```

Configure pseudowire 1 (vpws-pw-1) as dynamic:

```
! 
xconnect group vpws
p2p pw1
interface TenGigE0/5/0/9.1
neighbor ipv4 1.1.1.1 pw-id 1
bandwidth 1000
pw-class vpws-pw-1
```

---

Configure Ethernet OAM

CFM - Sample Configuration Workflow
### Sample Configuration on PE1

Configure pseudowire 2(vpws-pw-2) as static:

```plaintext
! 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
  !
!```

### Sample Configuration on PE2

Configure pseudowire 2(vpws-pw-2) as static:

```plaintext
! 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

```plaintext
l2vpn
xconnect group xc1
p2p pw2
interface TenGigE0/5/0/9
interface HundredGigE0/5/0/0
  !
!```

### Sample Configuration on PE2

```plaintext
l2vpn
xconnect group xc1
p2p pw2
interface TenGigE0/5/0/9
interface HundredGigE0/5/0/0
  !
!```

```plaintext
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
  !
```

```plaintext
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
  !
```

```plaintext
domain MD2 level 2 id null
  !
```

```plaintext
domain MD2 level 2 id null
  !
```

```plaintext
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
  !
```
Configure Ethernet OAM

### EFD Configuration: Examples

This example shows how to enable EFD:

```
RP/0/RP0:hostname# configure
RP/0/RP0:hostname(config)# ethernet cfm
RP/0/RP0:hostname(config-cfm)# domain D1 level 1 id null
```

---

<table>
<thead>
<tr>
<th>Sample Configuration on PE1</th>
<th>Sample Configuration on PE2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ais transmission interval 1s cos 0</td>
<td>ais transmission interval 1s cos 0</td>
</tr>
<tr>
<td>log ais</td>
<td>log ais</td>
</tr>
<tr>
<td>log continuity-check errors</td>
<td>log continuity-check errors</td>
</tr>
<tr>
<td>log crosscheck errors</td>
<td>log crosscheck errors</td>
</tr>
<tr>
<td>log continuity-check mep changes</td>
<td>log continuity-check mep changes</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
<tr>
<td>!</td>
<td>!</td>
</tr>
</tbody>
</table>

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
!
!
!

---

Configure Ethernet OAM

69
RP/0/RP0:hostname(config-cfm-dmn)# service S1 down-meps id number 1
RP/0/RP0:hostname(config-cfm-dmn-svc)# efd

This example shows how to enable EFD logging:

RP/0/RP0:hostname# configure
RP/0/RP0:hostname(config)# ethernet cfm
RP/0/RP0:hostname(config-cfm)# domain D1 level 1 id null
RP/0/RP0:hostname(config-cfm-dmn)# service S1 down-meps id number 1
RP/0/RP0:hostname(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/RP0:hostname# 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/RP0:hostname# 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
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 12: 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/RP0:hostname# configure
!
! Configure the Local UNI EFPs
!
RP/0/RP0:hostname(config)# interface TenGigE0/0/0/0.0 l2transport
RP/0/RP0:hostname(config-subif)# encapsulation dot1q 1-20
RP/0/RP0:hostname(config-subif)# exit
RP/0/RP0:hostname(config)# interface TenGigE0/0/0/1.1 l2transport
RP/0/RP0:hostname(config-subif)# encapsulation dot1q 1-20
RP/0/RP0:hostname(config-subif)# exit
!
! Create the EVC
!
RP/0/RP0:hostname(config)# l2vpn
RP/0/RP0:hostname(config-l2vpn)# bridge group BG1
RP/0/RP0:hostname(config-l2vpn-bd)# interface TenGigE0/0/0/0.0
RP/0/RP0:hostname(config-l2vpn-bd)# interface TenGigE0/0/0/1.1
RP/0/RP0:hostname(config-l2vpn-bd)# exit
RP/0/RP0:hostname(config-l2vpn-bd)# exit
RP/0/RP0:hostname(config-l2vpn)# exit
!
! Configure Ethernet CFM
!
RP/0/RP0:hostname(config)# ethernet cfm
RP/0/RP0:hostname(config-cfm)# domain GLOBAL level 5
RP/0/RP0:hostname(config-cfm-dmn)# service CustomerA bridge group BG1 bridge-domain BD1
RP/0/RP0:hostname(config-cfm-dmn-svc)# continuity-check interval 100ms
RP/0/RP0:hostname(config-cfm-dmn-svc)# mep crosscheck mep-id 22
RP/0/RP0:hostname(config-cfm-dmn-svc)# mep crosscheck mep-id 11
RP/0/RP0:hostname(config-cfm-dmn-svc)# exit
```
Configure Ethernet OAM

Configuration Example for Ethernet LMI

```
RP/0/RP0:hostname(config-cfm-dmn)# exit
RP/0/RP0:hostname(config-cfm)# exit
!
! Configure EFPs as CFM MEPs
!
RP/0/RP0:hostname(config)# interface TenGigE0/0/0/0.0 l2transport
RP/0/RP0:hostname(config-subif)# ethernet cfm
RP/0/RP0:hostname(config-if-cfm)# mep domain GLOBAL service CustomerA mep-id 22
RP/0/RP0:hostname(config-if-cfm)# exit
RP/0/RP0:hostname(config-subif)# exit
!
! Configure the Local UNI Name
!
RP/0/RP0:hostname(config)# interface TenGigE 0/0/0/0
RP/0/RP0:hostname(config-if)# ethernet uni id PE1-CustA-Slot0-Port0
RP/0/RP0:hostname(config-if)# exit
!
! Enable E-LMI on the Local UNI Physical Interface
!
RP/0/RP0:hostname(config)# interface TenGigE 0/0/0/0
RP/0/RP0:hostname(config-if)# ethernet lmi
RP/0/RP0:hostname(config-if)# exit
RP/0/RP0:hostname(config)# commit
```