Carrier Ethernet Configuration Guide, Cisco IOS Release 12.2SX

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

Using Ethernet Operations Administration and Maintenance

Ethernet Operations, Administration, and Maintenance (OAM) is a protocol for installing, monitoring, and troubleshooting Ethernet metropolitan-area networks (MANs) and Ethernet WANs. It relies on a new, optional sublayer in the data link layer of the Open Systems Interconnection (OSI) model. The OAM features covered by this protocol are Discovery, Link Monitoring, Remote Fault Detection, Remote Loopback, and Cisco Proprietary Extensions.

The advent of Ethernet as a MAN and WAN technology has emphasized the necessity for integrated management for larger deployments. For Ethernet to extend into public MANs and WANs, it must be equipped with a new set of requirements on Ethernet's traditional operations, which had been centered on enterprise networks only. The expansion of Ethernet technology into the domain of service providers, where networks are substantially larger and more complex than enterprise networks and the user-base is wider, makes operational management of link uptime crucial.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Information About Using Ethernet Operations Administration and Maintenance

Ethernet OAM

Ethernet OAM is a protocol for installing, monitoring, and troubleshooting metro Ethernet networks and Ethernet WANs. It relies on a new, optional sublayer in the data link layer of the OSI model. Ethernet OAM can be implemented on any full-duplex point-to-point or emulated point-to-point Ethernet link. A system-wide implementation is not required; OAM can be deployed for part of a system; that is, on particular interfaces.

Normal link operation does not require Ethernet OAM. OAM frames, called OAM protocol data units (PDUs), use the slow protocol destination MAC address 0180.c200.0002. They are intercepted by the MAC sublayer and cannot propagate beyond a single hop within an Ethernet network.

Ethernet OAM is a relatively slow protocol with modest bandwidth requirements. The frame transmission rate is limited to a maximum of 10 frames per second; therefore, the impact of OAM on normal operations is negligible. However, when link monitoring is enabled, the CPU must poll error counters frequently. In this case, the required CPU cycles will be proportional to the number of interfaces that have to be polled.

Two major components, the OAM client and the OAM sublayer, make up Ethernet OAM. The following two sections describe these components.

OAM Client

The OAM client is responsible for establishing and managing Ethernet OAM on a link. The OAM client also enables and configures the OAM sublayer. During the OAM discovery phase, the OAM client monitors OAM PDUs received from the remote peer and enables OAM functionality on the link based on local and remote state as well as configuration settings. Beyond the discovery phase (at steady state), the OAM client is responsible for managing the rules of response to OAM PDUs and managing the OAM remote loopback mode.

OAM Sublayer

The OAM sublayer presents two standard IEEE 802.3 MAC service interfaces: one facing toward the superior sublayers, which include the MAC client (or link aggregation), and the other interface facing toward the subordinate MAC control sublayer. The OAM sublayer provides a dedicated interface for passing OAM control information and OAM PDUs to and from a client.

The OAM sublayer is made up of three components: control block, multiplexer, and packet parser (p-parser). Each component is described in the following sections.

Control Block

The control block provides the interface between the OAM client and other blocks internal to the OAM sublayer. The control block incorporates the discovery process, which detects the existence and capabilities of remote OAM peers. It also includes the transmit process that governs the transmission of OAM PDUs to the multiplexer and a set of rules that govern the receipt of OAM PDUs from the p-parser.

Multiplexer

The multiplexer manages frames generated (or relayed) from the MAC client, control block, and p-parser. The multiplexer passes through frames generated by the MAC client untouched. It passes OAM PDUs generated by the control block to the subordinate sublayer; for example, the MAC sublayer. Similarly, the multiplexer passes loopback frames from the p-parser to the same subordinate sublayer when the interface is in OAM remote loopback mode.

P-Parser

The p-parser classifies frames as OAM PDUs, MAC client frames, or loopback frames and then dispatches each class to the appropriate entity. OAM PDUs are sent to the control block. MAC client frames are passed to the superior sublayer. Loopback frames are dispatched to the multiplexer.

Benefits of Ethernet OAM

Ethernet OAM provides the following benefits:

- Competitive advantage for service providers
- Standardized mechanism to monitor the health of a link and perform diagnostics

Cisco IOS Implementation of Ethernet OAM

The Cisco IOS implementation of Ethernet OAM consists of the Ethernet OAM shim and the Ethernet OAM module.

The Ethernet OAM shim is a thin layer that connects the Ethernet OAM module and the platform code. It is implemented in the platform code (driver). The shim also communicates port state and error conditions to the Ethernet OAM module via control signals.

The Ethernet OAM module, implemented within the control plane, handles the OAM client as well as control block functionality of the OAM sublayer. This module interacts with the command-line interface (CLI) and Simple Network Management Protocol (SNMP)/programmatic interface via control signals. In addition, this module interacts with the Ethernet OAM shim through OAM PDU flows.

OAM Features

The OAM features as defined by IEEE 802.3ah, *Ethernet in the First Mile* , are discovery, Link Monitoring, Remote Fault Detection, Remote Loopback, and Cisco Proprietary Extensions.

Discovery

Discovery is the first phase of Ethernet OAM and it identifies the devices in the network and their OAM capabilities. Discovery uses information OAM PDUs. During the discovery phase, the following information is advertised within periodic information OAM PDUs:

- OAM mode--Conveyed to the remote OAM entity. The mode can be either active or passive and can be used to determine device functionality.

- OAM configuration (capabilities)--Advertises the capabilities of the local OAM entity. With this information a peer can determine what functions are supported and accessible; for example, loopback capability.
- OAM PDU configuration--Includes the maximum OAM PDU size for receipt and delivery. This information along with the rate limiting of 10 frames per second can be used to limit the bandwidth allocated to OAM traffic.
- Platform identity--A combination of an organization unique identifier (OUI) and 32-bits of vendor-specific information. OUI allocation, controlled by the IEEE, is typically the first three bytes of a MAC address.

Discovery includes an optional phase in which the local station can accept or reject the configuration of the peer OAM entity. For example, a node may require that its partner support loopback capability to be accepted into the management network. These policy decisions may be implemented as vendor-specific extensions.

Link Monitoring

Link monitoring in Ethernet OAM detects and indicates link faults under a variety of conditions. Link monitoring uses the event notification OAM PDU and sends events to the remote OAM entity when there are problems detected on the link. The error events include the following:

- Error Symbol Period (error symbols per second)--The number of symbol errors that occurred during a specified period exceeded a threshold. These errors are coding symbol errors.
- Error Frame (error frames per second)--The number of frame errors detected during a specified period exceeded a threshold.
- Error Frame Period (error frames per *n* frames)--The number of frame errors within the last n frames has exceeded a threshold.
- Error Frame Seconds Summary (error seconds per *m* seconds)--The number of error seconds (1-second intervals with at least one frame error) within the last m seconds has exceeded a threshold.

Since IEEE 802.3ah OAM does not provide a guaranteed delivery of any OAM PDU, the event notification OAM PDU may be sent multiple times to reduce the probability of a lost notification. A sequence number is used to recognize duplicate events.

Remote Failure Indication

Faults in Ethernet connectivity that are caused by slowly deteriorating quality are difficult to detect. Ethernet OAM provides a mechanism for an OAM entity to convey these failure conditions to its peer via specific flags in the OAM PDU. The following failure conditions can be communicated:

- Link Fault--Loss of signal is detected by the receiver; for instance, the peer's laser is malfunctioning. A link fault is sent once per second in the information OAM PDU. Link fault applies only when the physical sublayer is capable of independently transmitting and receiving signals.
- Dying Gasp--An unrecoverable condition has occurred; for example, a power failure. This type of condition is vendor specific. A notification about the condition may be sent immediately and continuously.
- Critical Event--An unspecified critical event has occurred. This type of event is vendor specific. A critical event may be sent immediately and continuously.

Remote Loopback

An OAM entity can put its remote peer into loopback mode using the loopback control OAM PDU. Loopback mode helps an administrator ensure the quality of links during installation or when troubleshooting. In loopback mode, every frame received is transmitted back on the same port except for

OAM PDUs and pause frames. The periodic exchange of OAM PDUs must continue during the loopback state to maintain the OAM session.

The loopback command is acknowledged by responding with an information OAM PDU with the loopback state indicated in the state field. This acknowledgement allows an administrator, for example, to estimate if a network segment can satisfy a service-level agreement. Acknowledgement makes it possible to test delay, jitter, and throughput.

When an interface is set to the remote loopback mode the interface no longer participates in any other Layer 2 or Layer 3 protocols; for example Spanning Tree Protocol (STP) or Open Shortest Path First (OSPF). The reason is that when two connected ports are in a loopback session, no frames other than the OAM PDUs are sent to the CPU for software processing. The non-OAM PDU frames are either looped back at the MAC level or discarded at the MAC level.

From a user’s perspective, an interface in loopback mode is in a link-up state.

Cisco Vendor-Specific Extensions

Ethernet OAM allows vendors to extend the protocol by allowing them to create their own type-length-value (TLV) fields.

OAM Messages

Ethernet OAM messages or OAM PDUs are standard length, untagged Ethernet frames within the normal frame length bounds of 64 to 1518 bytes. The maximum OAM PDU frame size exchanged between two peers is negotiated during the discovery phase.

OAM PDUs always have the destination address of slow protocols (0180.c200.0002) and an Ethertype of 8809. OAM PDUs do not go beyond a single hop and have a hard-set maximum transmission rate of 10 OAM PDUs per second. Some OAM PDU types may be transmitted multiple times to increase the likelihood that they will be successfully received on a deteriorating link.

Four types of OAM messages are supported:

- Information OAM PDU--A variable-length OAM PDU that is used for discovery. This OAM PDU includes local, remote, and organization-specific information.
- Event notification OAM PDU--A variable-length OAM PDU that is used for link monitoring. This type of OAM PDU may be transmitted multiple times to increase the chance of a successful receipt; for example, in the case of high-bit errors. Event notification OAM PDUs also may include a time stamp when generated.
- Loopback control OAM PDU--An OAM PDU fixed at 64 bytes in length that is used to enable or disable the remote loopback command.
- Vendor-specific OAM PDU--A variable-length OAM PDU that allows the addition of vendor-specific extensions to OAM.

IEEE 802.3ah Link Fault RFI Support

The IEEE 802.3ah Link Fault RFI Support feature provides a per-port configurable option that moves a port into a blocking state when an OAM PDU control request packet is received with the Link Fault Status flag set. In the blocking state, the port can continue to receive OAM PDUs, detect remote link status, and automatically recover when the remote link becomes operational. When an OAM PDU is received with the Link Fault Status flag set to zero or FALSE, the port is enabled and all VLANs configured on the port are set to “forwarding.”

Note If you configure the Ethernet OAM timeout period to be the minimum allowable value of 2 seconds, the Ethernet OAM session may be dropped briefly when the port transitions from blocked to unblocked. This action will not occur by default; the default timeout value is 5 seconds.

Before the release of the IEEE 802.3ah Link Fault RFI Support feature, when an OAM PDU control request packet was received with the Link Fault Status flag set, one of three actions was taken:

- The port was put in the error-disable state, meaning that the port did not send or receive packets, including Bridge Protocol Data Units (BPDU) packets. In the error-disable state, a link can automatically recover after the error-disable timeout period but cannot recover automatically when the remote link becomes operational.
- A warning message was displayed or logged, and the port remained operational.
- The Link Fault Status flag was ignored.

A new keyword, **error-block-interface**, for the CLI command **ethernet oam remote-failure action** is introduced with the IEEE 802.3ah Link Fault RFI Support feature. For detailed information about this command, see the *Cisco IOS Carrier Ethernet Command Reference*.

Ethernet Connectivity Fault Management

Ethernet connectivity fault management (CFM) is an end-to-end per-service-instance Ethernet layer OAM protocol that includes proactive connectivity monitoring, fault verification, and fault isolation. End to end can be provider edge (PE) to PE or customer edge (CE) to CE. Per service instance means per VLAN.

For more information about Ethernet CFM, see Ethernet Connectivity Fault Management .

High Availability Features Supported by 802.3ah

In access and service provider networks using Ethernet technology, High Availability (HA) is a requirement, especially on Ethernet OAM components that manage Ethernet virtual circuit (EVC) connectivity. End-to-end connectivity status information is critical and must be maintained on a hot standby Route Processor (RP) (a standby RP that has the same software image as the active RP and supports synchronization of line card, protocol, and application state information between RPs for supported features and protocols). End-to-end connectivity status is maintained on the CE, PE, and access aggregation PE (uPE) network nodes based on information received by protocols such as Ethernet LMI, CFM, and 802.3ah. This status information is used to either stop traffic or switch to backup paths when an EVC is down. The Non-Stop Forwarding/Stateful Switchover (NSF/SSO) and In Service Software Upgrade (ISSU) support enhancements are introduced and enabled automatically during configuration of the Cisco 7600 router. Metro Ethernet clients (E-LMI, CFM, 802.3ah) maintain configuration data and dynamic data, which is learned through protocols. Every transaction involves either accessing or updating data among the various databases. If the databases are synchronized across active and standby modules, the RPs are transparent to clients.

Cisco IOS infrastructure provides various component application program interfaces (APIs) for clients that are helpful in maintaining a hot standby RP. Metro Ethernet HA clients (Ethernet LMI, HA/ISSU, CFM HA/ISSU, 802.3ah HA/ISSU) interact with these components, update the databases, and trigger necessary events to other components.

- Benefits of 802.3ah HA, page 7

Benefits of 802.3ah HA

- Elimination of network downtime for Cisco IOS software image upgrades, resulting in higher availability
- Elimination of resource scheduling challenges associated with planned outages and late night maintenance windows
- Accelerated deployment of new services and applications and faster implementation of new features, hardware, and fixes due to the elimination of network downtime during upgrades
- Reduced operating costs due to outages while delivering higher service levels due to the elimination of network downtime during upgrades

NSF SSO Support in 802.3ah OAM

The redundancy configurations SSO and NSF are both supported in Ethernet OAM and are automatically enabled. A switchover from an active to a standby RP occurs when the active RP fails, is removed from the networking device, or is manually taken down for maintenance. NSF interoperates with the SSO feature to minimize network downtime following a switchover. The primary function of Cisco NSF is to continue forwarding IP packets following an RP switchover.

For detailed information about the SSO feature, see the “Stateful Switchover” chapter of the *Cisco IOS High Availability Configuration Guide*. For detailed information about the NSF feature, see the “Cisco Nonstop Forwarding” chapter of the *Cisco IOS High Availability Configuration Guide.*

ISSU Support in 802.3ah OAM

ISSU allows you to perform a Cisco IOS software upgrade or downgrade without disrupting packet flow. ISSU is automatically enabled in 802.3ah. OAM performs a bulk update and a runtime update of the continuity check database to the standby RP, including adding, deleting, or updating a row. This checkpoint data requires ISSU capability to transform messages from one release to another. All the components that perform active RP to standby RP updates using messages require ISSU support.

ISSU lowers the impact that planned maintenance activities have on network availability by allowing software changes while the system is in service. For detailed information about ISSU, see the “Cisco OS In Service Software Upgrade Process” chapter of the *Cisco IOS High Availability Configuration Guide*.

How to Set Up and Configure Ethernet Operations Administration and Maintenance

Enabling Ethernet OAM on an Interface

Ethernet OAM is by default disabled on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet oam** [**max-rate** *oampdus* | **min-rate** *num-seconds*| **mode** {**active** | **passive**} | **timeout** *seconds*]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 3/8`	Specifies an interface and enters interface configuration mode.
Step 4	**ethernet oam** [**max-rate** *oampdus* | **min-rate** *num-seconds*| **mode** {**active** | **passive**} | **timeout** *seconds*] **Example:** `Router(config-if)# ethernet oam`	Enables Ethernet OAM.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.

Disabling and Enabling a Link Monitoring Session

Link monitoring is enabled by default when you enable Ethernet OAM. Perform these tasks to disable and enable link monitoring sessions:

- Disabling a Link Monitoring Session, page 9
- Enabling a Link Monitoring Session, page 10

Disabling a Link Monitoring Session

Perform this task to disable a link monitoring session.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet oam** [**max-rate** *oampdus* | **min-rate** *num-seconds*| **mode** {**active** | **passive**} | **timeout** *seconds*]
5. **no ethernet oam link-monitor supported**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitEthernet 3/8`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ethernet oam** [**max-rate** *oampdus* | **min-rate** *num-seconds*| **mode** {**active** | **passive**} | **timeout** *seconds*] **Example:** `Router(config-if)# ethernet oam`	Enables Ethernet OAM.
Step 5	**no ethernet oam link-monitor supported** **Example:** `Router(config-if)# no ethernet oam link-monitor supported`	Disables link monitoring on the interface.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.

Enabling a Link Monitoring Session

Perform this task to reenable a link monitoring session after it was previously disabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet oam link-monitor supported**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitEthernet 3/8`	Specifies an interface and enters interface configuration mode.
Step 4	**ethernet oam link-monitor supported** **Example:** `Router(config-if)# ethernet oam link-monitor supported`	Enables link monitoring on the interface.
Step 5	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.

Stopping and Starting Link Monitoring Operations

Link monitoring operations start automatically when Ethernet OAM is enabled on an interface. When link monitoring operations are stopped, the interface does not actively send or receive event notification OAM PDUs. The tasks in this section describe how to stop and start link monitoring operations.

- Stopping Link Monitoring Operations, page 11
- Starting Link Monitoring Operations, page 13

Stopping Link Monitoring Operations

Perform this task to stop link monitoring operations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet oam** [**max-rate** *oampdus* | **min-rate** *num-seconds*| **mode** {**active** | **passive**} | **timeout** *seconds*]
5. **no ethernet oam link-monitor on**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 3/8`	Specifies an interface and enters interface configuration mode.
Step 4	**ethernet oam** [**max-rate** *oampdus* | **min-rate** *num-seconds*| **mode** {**active** | **passive**} | **timeout** *seconds*] **Example:** `Router(config-if)# ethernet oam`	Enables Ethernet OAM.
Step 5	**no ethernet oam link-monitor on** **Example:** `Router(config-if)# no ethernet oam link-monitor on`	Stops link monitoring operations.

	Command or Action	Purpose
Step 6	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.

Starting Link Monitoring Operations

Perform this task to start link monitoring operations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet oam link-monitor on**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 3/8`	Specifies an interface and enters interface configuration mode.
Step 4	**ethernet oam link-monitor on** **Example:** `Router(config-if)# ethernet oam link-monitor on`	Starts link monitoring operations.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.

Configuring Link Monitoring Options

Perform this optional task to specify link monitoring options. Steps 4 through 10 can be performed in any sequence.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet oam** [**max-rate** *oampdus* | **min-rate** *num-seconds*| **mode** {**active** | **passive**} | **timeout** *seconds*]
5. **ethernet oam link-monitor high-threshold action error-disable-interface**
6. **ethernet oam link-monitor frame** {**threshold** {**high** {**none** | *high-frames*} | **low** *low-frames*} | **window** *milliseconds*}
7. **ethernet oam link-monitor frame-period** {**threshold** {**high** {**none** | *high-frames*} | **low** *low-frames*} | **window** *frames*}
8. **ethernet oam link-monitor frame-seconds** {**threshold** {**high** {**none** | *high-frames*} | **low** *low-frames*} | **window** *milliseconds*}
9. **ethernet oam link-monitor receive-crc** {**threshold** {**high** {*high-frames* | **none**} | **low** *low-frames*} | **window** *milliseconds*}
10. **ethernet oam link-monitor transmit-crc** {**threshold** {**high** {*high-frames* | **none**} | **low** *low-frames*} | **window** *milliseconds*}
11. **ethernet oam link-monitor symbol-period** {**threshold** {**high** {**none** | *high-symbols*} | **low** *low-symbols*} | **window** *symbols*}
12. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface gigabitEthernet 3/8`	Identifies the interface and enters interface configuration mode.
Step 4	**ethernet oam** [**max-rate** *oampdus* | **min-rate** *num-seconds*| **mode** {**active** | **passive**} | **timeout** *seconds*] **Example:** `Router(config-if)# ethernet oam`	Enables Ethernet OAM.
Step 5	**ethernet oam link-monitor high-threshold action error-disable-interface** **Example:** `Router(config-if)# ethernet oam link-monitor high-threshold action error-disable-interface`	Configures an error-disable function on an Ethernet OAM interface when a high threshold for an error is exceeded.
Step 6	**ethernet oam link-monitor frame** {**threshold** {**high** {**none** | *high-frames*} | **low** *low-frames*} | **window** *milliseconds*} **Example:** `Router(config-if)# ethernet oam link-monitor frame window 399`	Configures a number for error frames that when reached triggers an action.
Step 7	**ethernet oam link-monitor frame-period** {**threshold** {**high** {**none** | *high-frames*} | **low** *low-frames*} | **window** *frames*} **Example:** `Router(config-if)# ethernet oam link-monitor frame-period threshold high 599`	Configures a number of frames to be polled. Frame period is a user-defined parameter.

	Command or Action	Purpose
Step 8	**ethernet oam link-monitor frame-seconds** {**threshold** {**high** {**none** | *high-frames*} | **low** *low-frames*} | **window** *milliseconds*} **Example:** `Router(config-if)# ethernet oam link-monitor frame-seconds window 699`	Configures a period of time in which error frames are counted.
Step 9	**ethernet oam link-monitor receive-crc** {**threshold** {**high** {*high-frames* | **none**} | **low** *low-frames*} | **window** *milliseconds*} **Example:** `Router(config-if)# ethernet oam link-monitor receive-crc window 99`	Configures an Ethernet OAM interface to monitor ingress frames with cyclic redundancy check (CRC) errors for a period of time.
Step 10	**ethernet oam link-monitor transmit-crc** {**threshold** {**high** {*high-frames* | **none**} | **low** *low-frames*} | **window** *milliseconds*} **Example:** `Router(config-if)# ethernet oam link-monitor transmit-crc threshold low 199`	Configures an Ethernet OAM interface to monitor egress frames with CRC errors for a period of time.
Step 11	**ethernet oam link-monitor symbol-period** {**threshold** {**high** {**none** | *high-symbols*} | **low** *low-symbols*} | **window** *symbols*} **Example:** `Router(config-if)# ethernet oam link-monitor symbol-period threshold high 299`	Configures a threshold or window for error symbols, in number of symbols.
Step 12	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.

Example

```
Router# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# interface gigabitEthernet 3/8
Router(config-if)#
Router(config-if)# ethernet oam

Router(config-if)# ethernet oam link-monitor high-threshold action error-disable-interface
Router(config-if)# ethernet oam link-monitor frame window 399
Router(config-if)# ethernet oam link-monitor frame-period threshold high 599
```

```
Router(config-if)# ethernet oam link-monitor frame-seconds window 699
Router(config-if)# ethernet oam link-monitor receive-crc window 99
Router(config-if)# ethernet oam link-monitor transmit-crc threshold low 199
Router(config-if)# ethernet oam link-monitor symbol-period threshold high 299
Router(config-if)# exit
Router# show running-config

Building configuration...
Current configuration : 5613 bytes
!
!
version 12.2
!
!
.
.
.
!
!
interface GigabitEthernet3/8
 no ip address
 ethernet oam link-monitor high-threshold action error-disable-interface
 ethernet oam link-monitor frame window 399
 ethernet oam link-monitor frame-period threshold high 599
 ethernet oam link-monitor frame-seconds window 699
 ethernet oam link-monitor receive-crc window 99
 ethernet oam link-monitor transmit-crc threshold low 199
 ethernet oam link-monitor symbol-period threshold high 299
 ethernet oam
```

Configuring Global Ethernet OAM Options Using a Template

Perform this task to create a template to use for configuring a common set of options on multiple Ethernet OAM interfaces. Steps 4 through 10 are optional and can be performed in any sequence. These steps may also be repeated to configure different options.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **template** *template-name*
4. **ethernet oam link-monitor receive-crc** {**threshold** {**high** {*high-frames* | **none**} | **low** *low-frames*} | **window** *milliseconds*}
5. **ethernet oam link-monitor transmit-crc** {**threshold** {**high** {*high-frames* | **none**} | **low** *low-frames*} | **window** *milliseconds*}
6. **ethernet oam link-monitor symbol-period** {**threshold** {**high** {**none** | *high-symbols*} | **low** *low-symbols*} | **window** *symbols*}
7. **ethernet oam link-monitor high-threshold action error-disable-interface**
8. **ethernet oam link-monitor frame** {**threshold** {**high** {**none** | *high-frames*} | **low** *low-frames*} | **window** *milliseconds*}
9. **ethernet oam link-monitor frame-period** {**threshold** {**high** {**none** | *high-frames*} | **low** *low-frames*} | **window** *frames*}
10. **ethernet oam link-monitor frame-seconds** {**threshold** {**high** {**none** | *high-frames*} | **low** *low-frames*} | **window** *milliseconds*}
11. **exit**
12. **interface** *type number*
13. **source template** *template-name*
14. **exit**
15. **exit**
16. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**template** *template-name* **Example:** `Router(config)# template oam-temp`	Configures a template and enters template configuration mode.

	Command or Action	Purpose
Step 4	**ethernet oam link-monitor receive-crc** {**threshold** {**high** {*high-frames* | **none**} | **low** *low-frames*} | **window** *milliseconds*} **Example:** `Router(config-template)# ethernet oam link-monitor receive-crc window 99`	Configures an Ethernet OAM interface to monitor ingress frames with CRC errors for a period of time.
Step 5	**ethernet oam link-monitor transmit-crc** {**threshold** {**high** {*high-frames* | **none**} | **low** *low-frames*} | **window** *milliseconds*} **Example:** `Router(config-template)# ethernet oam link-monitor transmit-crc threshold low 199`	Configures an Ethernet OAM interface to monitor egress frames with CRC errors for a period of time.
Step 6	**ethernet oam link-monitor symbol-period** {**threshold** {**high** {**none** | *high-symbols*} | **low** *low-symbols*} | **window** *symbols*} **Example:** `Router(config-template)# ethernet oam link-monitor symbol-period threshold high 299`	Configures a threshold or window for error symbols, in number of symbols.
Step 7	**ethernet oam link-monitor high-threshold action error-disable-interface** **Example:** `Router(config-template)# ethernet oam link-monitor high-threshold action error-disable-interface`	Configures an error-disable function on an Ethernet OAM interface when a high threshold for an error is exceeded.
Step 8	**ethernet oam link-monitor frame** {**threshold** {**high** {**none** | *high-frames*} | **low** *low-frames*} | **window** *milliseconds*} **Example:** `Router(config-template)# ethernet oam link-monitor frame window 399`	Configures a number for error frames that when reached triggers an action.
Step 9	**ethernet oam link-monitor frame-period** {**threshold** {**high** {**none** | *high-frames*} | **low** *low-frames*} | **window** *frames*} **Example:** `Router(config-template)# ethernet oam link-monitor frame-period threshold high 599`	Configures a number of frames to be polled. Frame period is a user-defined parameter.

	Command or Action	Purpose
Step 10	**ethernet oam link-monitor frame-seconds** {**threshold** {**high** {**none** | *high-frames*} | **low** *low-frames*} | **window** *milliseconds*} **Example:** `Router(config-template)# ethernet oam link-monitor frame-seconds window 699`	Configures a period of time in which error frames are counted.
Step 11	**exit** **Example:** `Router(config-template)# exit`	Returns the CLI to global configuration mode.
Step 12	**interface** *type number* **Example:** `Router(config)# interface gigabitEthernet 3/8`	Identifies the interface on which to use the template and enters interface configuration mode.
Step 13	**source template** *template-name* **Example:** `Router(config-if)# source template oam-temp`	Applies to the interface the options configured in the template.
Step 14	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.
Step 15	**exit** **Example:** `Router(config)# exit`	Returns the CLI to privileged EXEC mode.
Step 16	**show running-config** **Example:** `Router# show running-config`	Displays the updated running configuration.

Configuring a Port for Link Fault RFI Support

Perform this task to put a port into a blocking state when an OAM PDU control request packet is received with the Link Fault Status flag set.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet oam remote-failure** {**critical-event** | **dying-gasp** | **link-fault**} **action** {**error-block-interface** | **error-disable-interface**}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet 1/2`	Enters interface configuration mode.
Step 4	**ethernet oam remote-failure** {**critical-event** | **dying-gasp** | **link-fault**} **action** {**error-block-interface** | **error-disable-interface**} **Example:** `Router(config-if)# ethernet oam remote-failure critical-event action error-block-interface`	Sets the interface to the blocking state when a critical event occurs.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.

Configuration Examples for Ethernet Operations Administration and Maintenance

The following example shows how to configure Ethernet OAM options using a template and overriding that configuration by configuring an interface. In this example, the network supports a Gigabit Ethernet interface between the customer edge device and provider edge device.

```
! Configure a global OAM template for both PE and CE configuration.
!
Router(config)# template oam
Router(config-template)# ethernet oam link-monitor symbol-period threshold low 10
Router(config-template)# ethernet oam link-monitor symbol-period threshold high 100
Router(config-template)# ethernet oam link-monitor frame window 100
Router(config-template)# ethernet oam link-monitor frame threshold low 10
Router(config-template)# ethernet oam link-monitor frame threshold high 100
Router(config-template)# ethernet oam link-monitor frame-period window 100
Router(config-template)# ethernet oam link-monitor frame-period threshold low 10
Router(config-template)# ethernet oam link-monitor frame-period threshold high 100
Router(config-template)# ethernet oam link-monitor frame-seconds window 1000
Router(config-template)# ethernet oam link-monitor frame-seconds threshold low 10
Router(config-template)# ethernet oam link-monitor frame-seconds threshold high 100
Router(config-template)# ethernet oam link-monitor receive-crc window 100
Router(config-template)# ethernet oam link-monitor receive-crc threshold high 100
Router(config-template)# ethernet oam link-monitor transmit-crc window 100
Router(config-template)# ethernet oam link-monitor transmit-crc threshold high 100
Router(config-template)# ethernet oam remote-failure dying-gasp action
error-disable-interface
Router(config-template)# exit
!
! Enable Ethernet OAM on the CE interface
!
Router(config)# interface gigabitethernet 4/1/1
Router(config-if)# ethernet oam
!
! Apply the global OAM template named "oam" to the interface.
!
Router(config-if)# source template oam
!
! Configure any interface-specific link monitoring commands to override the template
configuration. The following example disables the high threshold link monitoring for
receive CRC errors.
!
Router(config-if)# ethernet oam link-monitor receive-crc threshold high none
!
! Enable Ethernet OAM on the PE interface
!
Router(config)# interface gigabitethernet 8/1/1
Router(config-if)# ethernet oam
!
! Apply the global OAM template named "oam" to the interface.
!
Router(config-if)# source template oam
```

The following examples show how to verify various Ethernet OAM configurations and activities.

Verifying an OAM Session

The following example shows that the local OAM client, Gigabit Ethernet interface Gi6/1/1, is in session with a remote client with MAC address 0012.7fa6.a700 and OUI 00000C, which is the OUI for Cisco. The remote client is in active mode and has established capabilities for link monitoring and remote loopback for the OAM session.

```
Router# show ethernet oam summary
Symbols:          * - Master Loopback State,  # - Slave Loopback State
Capability codes: L - Link Monitor,  R - Remote Loopback
                  U - Unidirection,  V - Variable Retrieval
  Local                        Remote
Interface       MAC Address    OUI    Mode    Capability
 Gi6/1/1       0012.7fa6.a700 00000C active      L R
```

Verifying OAM Discovery Status

The following example shows how to verify OAM discovery status of a local client and a remote peer:

```
Router# show ethernet oam discovery interface gigabitethernet6/1/1
GigabitEthernet6/1/1
Local client
------------
  Administrative configurations:
    Mode:              active
    Unidirection:      not supported
    Link monitor:      supported (on)
    Remote loopback:   not supported
    MIB retrieval:     not supported
    Mtu size:          1500
  Operational status:
Port status:       operational
    Loopback status:   no loopback
    PDU permission:    any
    PDU revision:      1
Remote client
-------------
  MAC address: 0030.96fd.6bfa
  Vendor(oui): 0x00 0x00 0x0C (cisco)
  Administrative configurations:
    Mode:              active
   Unidirection:      not supported
   Link monitor:      supported
   Remote loopback:   not supported
   MIB retrieval:     not supported
   Mtu size:          1500
```

Verifying Information OAMPDU and Fault Statistics

The following example shows how to verify statistics for information OAM PDUs and local and remote faults:

```
Router# show ethernet oam statistics interface gigabitethernet6/1/1
GigabitEthernet6/1/1
Counters:
---------
Information OAMPDU Tx                   : 588806
Information OAMPDU Rx                   : 988
Unique Event Notification OAMPDU Tx     : 0
Unique Event Notification OAMPDU Rx     : 0
Duplicate Event Notification OAMPDU TX  : 0
Duplicate Event Notification OAMPDU RX  : 0
Loopback Control OAMPDU Tx              : 1
Loopback Control OAMPDU Rx              : 0
```

```
Variable Request OAMPDU Tx              : 0
Variable Request OAMPDU Rx              : 0
Variable Response OAMPDU Tx             : 0
Variable Response OAMPDU Rx             : 0
Cisco OAMPDU Tx                         : 4
Cisco OAMPDU Rx                         : 0
Unsupported OAMPDU Tx                   : 0
Unsupported OAMPDU Rx                   : 0
Frames Lost due to OAM                  : 0
Local Faults:
-------------
0 Link Fault records
2 Dying Gasp records
Total dying gasps       : 4
Time stamp              : 00:30:39
Total dying gasps       : 3
Time stamp              : 00:32:39
0 Critical Event records
Remote Faults:
--------------
0 Link Fault records
0 Dying Gasp records
0 Critical Event records
Local event logs:
-----------------
0 Errored Symbol Period records
0 Errored Frame records
0 Errored Frame Period records
0 Errored Frame Second records
Remote event logs:
------------------
0 Errored Symbol Period records
0 Errored Frame records
0 Errored Frame Period records
0 Errored Frame Second records
```

Verifying Link Monitoring Configuration and Status

The following example shows how to verify link monitoring configuration and status on the local client. The highlighted Status field in the example shows that link monitoring status is supported and enabled (on).

```
Router# show ethernet oam status interface gigabitethernet6/1/1
GigabitEthernet6/1/1
General
-------
  Mode:                  active
  PDU max rate:          10 packets per second
  PDU min rate:          1 packet per 1 second
  Link timeout:          5 seconds
  High threshold action: no action
Link Monitoring
---------------
  Status: supported (on)
  Symbol Period Error
    Window:              1 million symbols
    Low threshold:       1 error symbol(s)
    High threshold:      none
  Frame Error
    Window:              10 x 100 milliseconds
    Low threshold:       1 error frame(s)
    High threshold:      none
Frame Period Error
    Window:              1 x 100,000 frames
    Low threshold:       1 error frame(s)
    High threshold:      none
  Frame Seconds Error
    Window:              600 x 100 milliseconds
    Low threshold:       1 error second(s)
    High threshold:      none
```

Verifying Status of a Remote OAM Client

The following example shows that the local client interface Gi6/1/1 is connected to a remote client. Note the values in the Mode and Capability fields.

```
Router# show ethernet oam summary
Symbols:          * - Master Loopback State,  # - Slave Loopback State
Capability codes: L - Link Monitor,  R - Remote Loopback
                  U - Unidirection,  V - Variable Retrieval
  Local                        Remote
Interface       MAC Address    OUI    Mode    Capability
 Gi6/1/1       0012.7fa6.a700 00000C active      L R
```

Additional References

Related Documents

Related Topic	Document Title
Ethernet CFM	Configuring Ethernet Connectivity Fault Management in a Service Provider Network" in the *Cisco OS Carrier Ethernet Configuration Guide*
Ethernet LMI	"Configuring Ethernet Local Management Interface" in the *Cisco IOS Carrier Ethernet Configuration Guide*
Configuring Ethernet LMI on a PE device	"Configuring Ethernet Local Management Interface at a Provider Edge" in the *Cisco IOS Carrier Ethernet Configuration Guide*
Carrier Ethernet commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Carrier Ethernet Command Reference*
Cisco IOS commands: master list of commands with complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS Master Commands List, All Releases

Standards

Standard	Title
IEEE Draft P802.3ah/D3.3	*Ethernet in the First Mile - Amendment*
IETF VPLS OAM	*L2VPN OAM Requirements and Framework*
ITU-T	ITU-T Y.1731 OAM Mechanisms for Ethernet-Based Networks

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Using Ethernet Operations Administration and Maintenance

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for Using Ethernet Operations, Administration, and Maintenance

Feature Name	Releases	Feature Information
Ethernet Operations, Administration, and Maintenance	12.2(33)SRA 12.2(33)SXH 12.4(15)T2 Cisco IOS XE 3.1.0SG	Ethernet OAM is a protocol for installing, monitoring, and troubleshooting metro Ethernet networks and Ethernet WANs. It relies on a new, optional sublayer in the data link layer of the OSI model. The OAM features covered by this protocol are Discovery, Link Monitoring, Remote Fault Detection, Remote Loopback, and Cisco Proprietary Extensions. The Ethernet Operations, Administration, and Maintenance feature was integrated into Cisco IOS Release 12.4(15)T. The Ethernet Operations, Administration, and Maintenance feature was integrated into Cisco IOS Release 12.2(33)SXH. The following commands were introduced or modified: **clear ethernet oam statistics, debug ethernet oam, ethernet oam, ethernet oam link-monitor frame, ethernet oam link-monitor frame-period, ethernet oam link-monitor frame-seconds, ethernet oam link-monitor high-threshold action, ethernet oam link-monitor on, ethernet oam link-monitor receive-crc, ethernet oam link-monitor supported, ethernet oam link-monitor symbol-period, ethernet oam link-monitor transmit-crc, ethernet oam remote-loopback, ethernet oam remote-loopback (interface), show ethernet oam discovery, show ethernet oam statistics, show ethernet oam status, show ethernet oam summary, source template (eoam), template (eoam)**.

Feature Name	Releases	Feature Information
IEEE 802.3ah Link Fault RFI Support	12.2(33)SXI	The IEEE 802.3ah Link Fault RFI Support feature provides a per-port configurable option that moves a port into a blocking state when an OAM PDU control request packet is received with the Link Fault Status flag set. In the blocking state, the port can continue to receive OAM PDUs, detect remote link status, and automatically recover when the remote link becomes operational. The following commands were introduced or modified: **ethernet oam remote-failure action**.
ISSU Support in 802.3ah OAM	12.2(33)SRD Cisco IOS XE 3.1.0SG	The ISSU Support in 802.3ah OAM feature allows software to be upgraded or downgraded without disrupting packet flow. In Cisco IOS Release 12.2(33)SRD, this feature was introduced on the Cisco 7600 series router.
NSF/SSO Support in 802.3ah OAM	12.2(33)SRD Cisco IOS XE 3.1.0SG	The NSF/SSO Support in 802.3ah OAM feature allows processes that support dual route processors in active and standby modes to continue forwarding packets following a switchover. In Cisco IOS Release 12.2(33)SRD, this feature was introduced on the Cisco 7600 series router.

Configuring IEEE Standard-Compliant Ethernet CFM in a Service Provider Network

Ethernet Connectivity Fault Management (CFM) is an end-to-end per-service Ethernet layer operations, administration, and maintenance (OAM) protocol. CFM includes proactive connectivity monitoring, fault verification, and fault isolation for large Ethernet metropolitan-area networks (MANs) and WANs.

The advent of Ethernet as a MAN and WAN technology imposes a new set of OAM requirements on Ethernet's traditional operations, which were centered on enterprise networks only. The expansion of Ethernet technology into the domain of service providers, where networks are substantially larger and more complex than enterprise networks and the user base is wider, makes operational management of link uptime crucial. More importantly, the timeliness in isolating and responding to a failure becomes mandatory for normal day-to-day operations, and OAM translates directly to the competitiveness of the service provider.

This document describes the implementation of IEEE 802.1ag Standard-Compliant CFM (IEEE CFM) in Cisco IOS software.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring IEEE Ethernet CFM in a Service Provider Network

- Network topology and network administration have been evaluated.
- Business and service policies have been established.
- Parser return codes (PRCs) have been implemented for all supported commands related to configuring CFM on a maintenance endpoint (MEP), maintenance intermediate point (MIP), level, service instance ID, cross-check timer, cross-check, and domain.
- To use Non-Stop Forwarding (NSF) and In Service Software Upgrade (ISSU), Stateful Switchover (SSO) must be configured and working properly.
- To deploy CFM and the Per VLAN Spanning Tree (PVST) Simulation feature, the Spanning Tree Protocol (STP) root switch must be inside the Multiple Spanning-Tree (MST) region.

Restrictions for Configuring IEEE Ethernet CFM in a Service Provider Network

- The IEEE CFM subsystem does not coexist in the same image as the Cisco pre-Standard CFM Draft 1 subsystem.
- IEEE CFM is supported on LAN cards. Linecards that do not support CFM will not boot up, but they display an error message.
- Unsupported line cards must be either removed or turned off.
- When physical ports are configured to a port channel on which CFM is configured, the following constraints apply:
    - Physical ports must allow use of the VLAN that is configured as part of the port channel’s CFM configuration.
    - CFM on secondary port channels is not supported.
    - CFM configuration on Fast EtherChannel (FEC) port channels is not supported.
- CFM is not fully supported on an MPLS provider edge (PE) device. There is no interaction between CFM and an EoMPLS pseudowire. CFM packets can be transparently passed like regular data packets only via pseudowire, with the following restrictions:
    - For Policy Feature Card (PFC)-based EoMPLS, which uses a Cisco Catalyst LAN card as the MPLS uplink port, a CFM packet can be transparently passed via an EoMPLS pseudowire like regular data packets. The EoMPLS endpoint interface, however, cannot be a MEP or a MIP, although a CFM MEP or MIP can be supported on regular Layer 2 switchport interfaces.
- High Availability (HA) feature support in CFM is platform dependent.
- CFM loopback messages will not be confined within a maintenance domain according to their maintenance level. The impact of not having CFM loopback messages confined to their maintenance levels occurs at these levels:
    - Architecture--CFM layering is violated for loopback messages.
    - Deployment--A user may potentially misconfigure a network and have loopback messages succeed.
    - Security--A malicious device that recognizes devices’ MAC addresses and levels may potentially explore a network topology that should be transparent.

- PVST simulation is not supported on blocked ports.

Information About Configuring IEEE Ethernet CFM in a Service Provider Network

IEEE CFM

IEEE CFM is an end-to-end per-service Ethernet layer OAM protocol that includes proactive connectivity monitoring, fault verification, and fault isolation. End to end can be PE to PE or customer edge to customer edge (CE to CE). A service can be identified as a service provider VLAN (S-VLAN) or an Ethernet virtual circuit (EVC) service.

Being an end-to-end technology is the distinction between CFM and other metro-Ethernet OAM protocols. For example, MPLS, ATM, and SONET OAM help in debugging Ethernet wires but are not always end to end. 802.3ah OAM is a single-hop and per-physical-wire protocol. It is not end to end or service aware. Ethernet Local Management Interface (E-LMI) is confined between the user-end provider edge (uPE) and CE and relies on CFM for reporting status of the metro-Ethernet network to the CE.

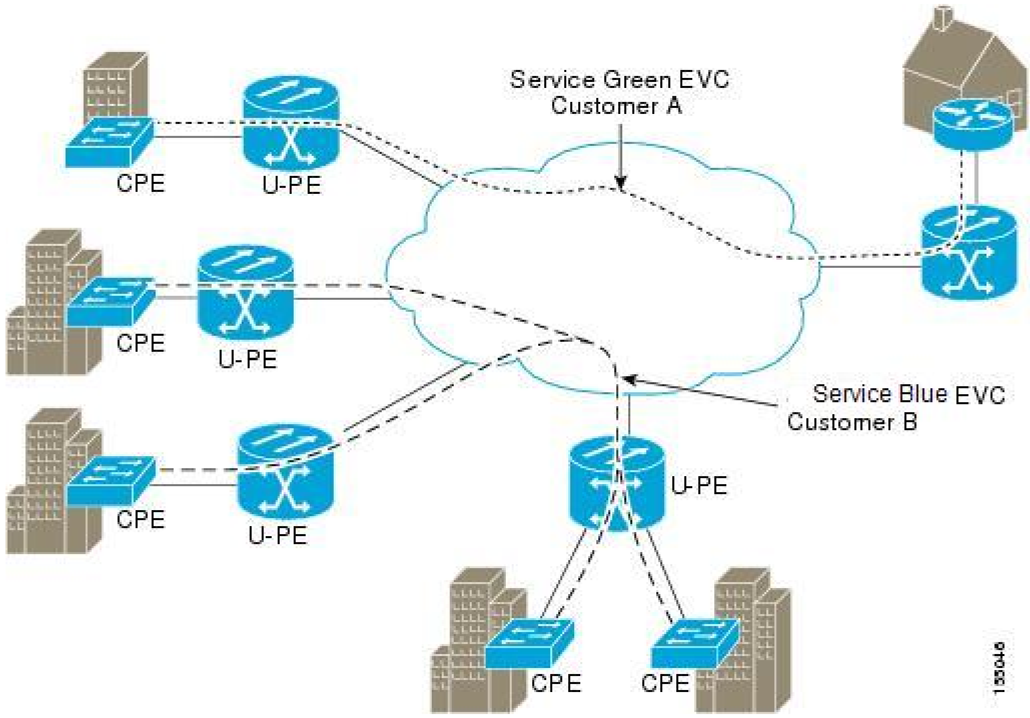
Troubleshooting carrier networks offering Ethernet Layer 2 services is challenging. Customers contract with service providers for end-to-end Ethernet service and service providers may subcontract with operators to provide equipment and networks. Compared to enterprise networks, where Ethernet traditionally has been implemented, these constituent networks belong to distinct organizations or departments, are substantially larger and more complex, and have a wider user base. Ethernet CFM provides a competitive advantage to service providers for which the operational management of link uptime and timeliness in isolating and responding to failures is crucial to daily operations.

Benefits of IEEE CFM

- End-to-end service-level OAM technology
- Reduced operating expense for service provider Ethernet networks
- Competitive advantage for service providers
- Support for both distribution and access network environments with Down (toward the wire) MEPs

Customer Service Instance

A customer service is an EVC, which is identified by the encapsulation VLAN within an Ethernet island, and is identified by a globally unique service ID. A customer service can be point-to-point or multipoint-to-multipoint. The figure below shows two customer services. Service Green is point to point; Service Blue is multipoint to multipoint.



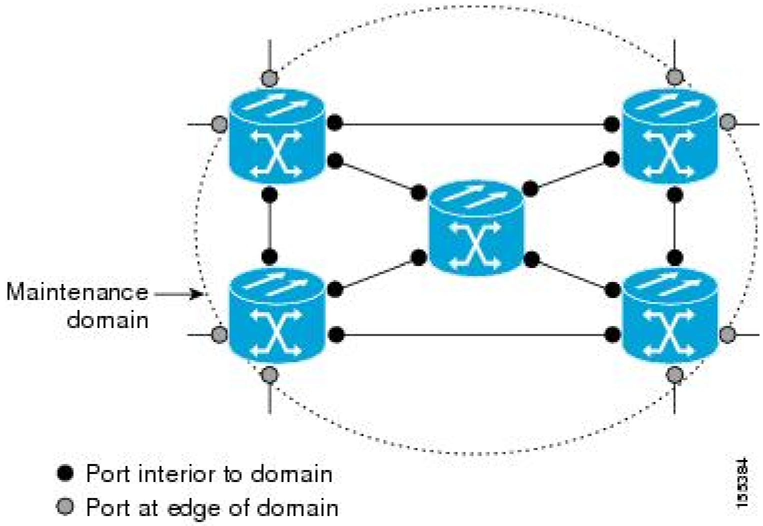
Maintenance Association

A maintenance association (MA) identifies a service that can be uniquely identified within a maintenance domain. There can be many MAs within a domain. The MA direction is specified when the MA is configured. The short MA name must be configured on a domain before MEPs can be configured. Configuring a MA is not required for devices that have only MIPs.

The CFM protocol runs for a specific MA.

Maintenance Domain

A maintenance domain is 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 ports internal to it and at its boundary. The figure below illustrates a typical maintenance domain.



A unique maintenance level in the range of 0 to 7 is assigned to each domain by a network administrator. Levels and domain names are useful for defining the hierarchical relationship that exists among domains. The hierarchical relationship of domains parallels the structure of customer, service provider, and operator. The larger the domain, the higher the level value. For example, a customer domain would be larger than an operator domain. The customer domain may have a maintenance level of 7 and the operator domain may have a maintenance level of 0. Typically, operators would have the smallest domains and customers the largest domains, with service provider domains between them in size. All levels of the hierarchy must operate together.

Domains should not intersect because intersecting would mean management by more than one entity, which is not allowed. Domains may nest or touch but when two domains nest, the outer domain must have a higher maintenance level than the domain nested within it. Nesting maintenance domains is useful in the business model where a service provider contracts with one or more operators to provide Ethernet service to a customer. Each operator would have its own maintenance domain and the service provider would define its domain--a superset of the operator domains. Furthermore, the customer has its own end-to-end domain, which is in turn a superset of the service provider domain. Maintenance levels of various nesting domains should be communicated among the administering organizations. For example, one approach would be to have the service provider assign maintenance levels to operators.

CFM exchanges messages and performs operations on a per-domain basis. For example, running CFM at the operator level does not allow discovery of the network by the higher provider and customer levels.

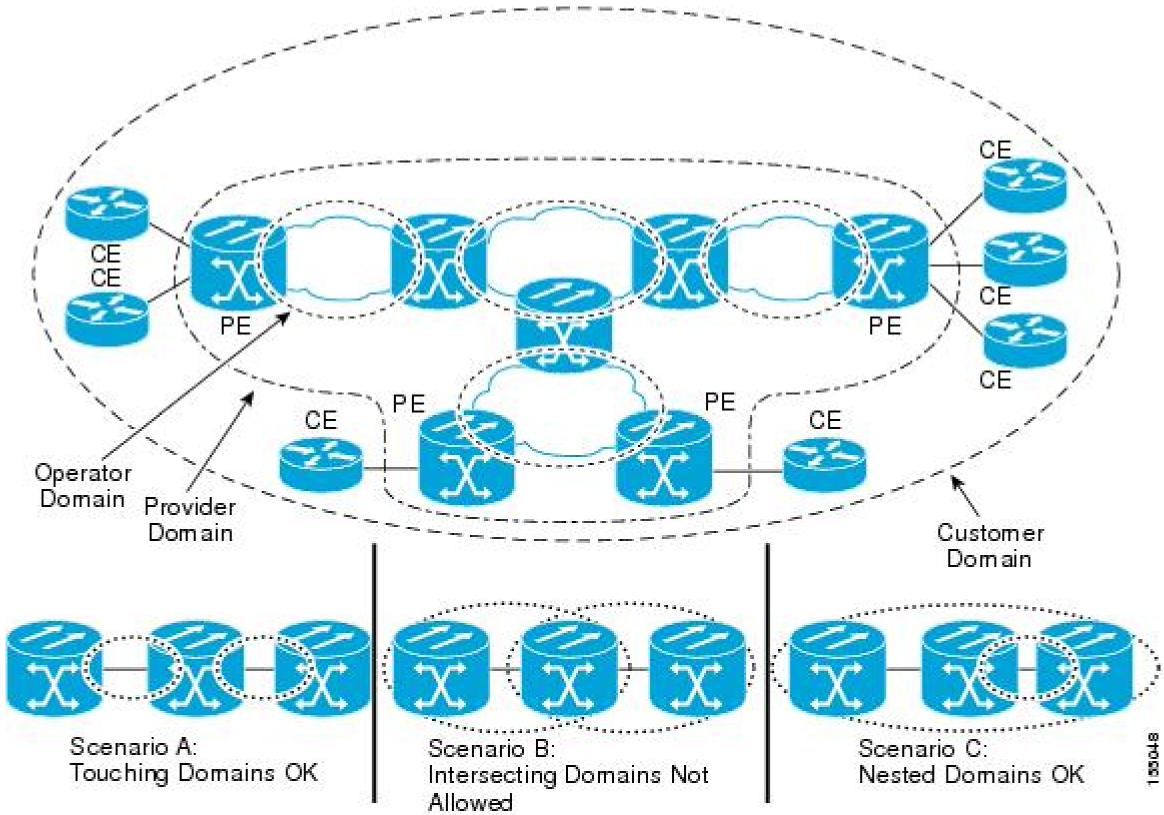
Network designers decide on domains and configurations.

The following characteristics of domains are supported:

- Name is a maximum of 154 characters
- Domain “null” is supported; the short maintenance association name is used as the identifier
- Domain configuration is not required for devices that have only MIPs
- Direction is specified when the maintenance association is configured
- Mix of Up (toward the bridge) and Down (toward the wire) MEPs is supported

A domain can be removed when all maintenance points within the domain have been removed and all remote MEP entries in the CCDB for the domain have been purged.

The figure below illustrates a hierarchy of operator, service provider, and customer domains and also illustrates touching, intersecting, and nested domains.



Maintenance Point

A maintenance point is a demarcation point on an interface or port that participates in CFM within a maintenance domain. Maintenance points on device ports act as filters that confine CFM frames within the bounds of a domain by dropping frames that do not belong to the correct level. Maintenance points must be explicitly configured on Cisco devices. Two classes of maintenance points exist, MEPs and MIPs.

- Maintenance Association Endpoints, page 36
- Maintenance Intermediate Points, page 37

Maintenance Association Endpoints

Maintenance association endpoints (MEPs) reside at the edge of a maintenance domain and confine CFM messages within the domain via the maintenance domain level. MEPs periodically transmit and receive continuity check messages (CCMs) from other MEPs within the domain. At the request of an administrator, linktrace and loopback messages can also be transmitted. MEPs are either “Up” (toward the bridge) or “Down” (toward the wire). The default direction is Up.

MEP supports multicast loopback and ping. When a multicast ping is done for a particular domain or service or vlan, all the related remote MEPs reply to the ping.

A port MEP supports a Down MEP with no VLAN and if a static remote MEP has not been detected, normal data traffic is stopped.

MEP configurations can be removed after all pending loopback and traceroute replies are removed and the service on the interface is set to transparent mode. To set the service to transparent mode, MIP filtering should not be configured.

Up MEPs

Up MEPs communicate through the Bridge Relay function and use the Bridge-Brain MAC address. An Up MEP performs the following functions:

- Sends and receives CFM frames at its level through the Bridge relay, not via the wire connected to the port on which the MEP is configured.
- Drops all CFM frames at its level (or lower level) that come from the direction of the wire.
- Processes all CFM frames at its level coming from the direction of the bridge.
- Drops all CFM frames at a lower level coming from the direction of the bridge.
- Transparently forwards all CFM frames at a higher level, independent of whether they come in from the bridge side or the wire side.
- If the port on which the Up MEP is configured is blocked by Spanning-Tree Protocol, the MEP can still transmit or receive CFM messages via the bridge function.

Down MEPs for Routed Ports and Switch Ports

Down MEPs communicate through the wire. They can be configured on routed ports and switch ports. A MIP configuration at a level higher than the level of a Down MEP is not required.

Down MEPs use the port MAC address. Down MEPs on port channels use the MAC address of the first member port. When port channel members change, the identities of Down MEPs do not have to change.

A Down MEP performs the following functions:

- Sends and receives CFM frames at its level via the wire connected to the port where the MEP is configured.
- Drops all CFM frames at its level (or at a lower level) that come from the direction of the bridge.
- Processes all CFM frames at its level coming from the direction of the wire.
- Drops all CFM frames at a lower level coming from the direction of the wire.
- If the port on which the Down MEP is configured is blocked by Spanning-Tree Protocol, the MEP can still transmit and receive CFM messages via the wire.
- Transparently forwards all CFM frames at a higher level, independent of whether they came in from the bridge or wire.

Maintenance Intermediate Points

Maintenance intermediate points (MIPs) are within a maintenance domain and catalog and forward information received from MEPs. MIPs are passive points that respond only to CFM linktrace and loopback messages. A MIP has only one level associated with it.

MIPs are defined as two MIP half functions (MHFs): An Up MHF that resides above the port filtering entities and a Down MHF that resides below the port filtering entities. The same configuration parameters and characteristics apply to both MHFs of a MIP, as follows:

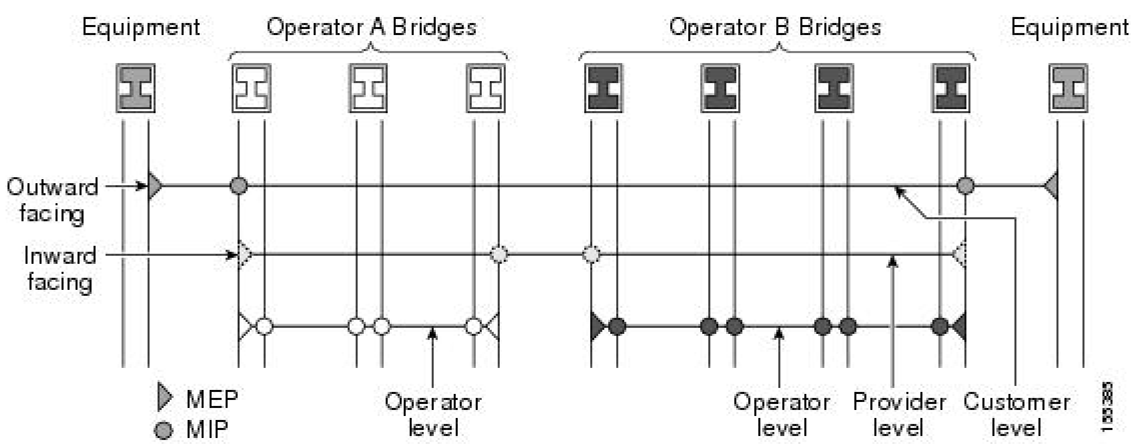
- Can be created manually or dynamically (auto MIPs)
- Dynamically created depending on configured policies at managed objects (MA, maintenance domain, or the default domain level)
- Manual MIPs can be created under an interface and under a service instance within an interface.
- Auto MIP commands can be issued globally or under a domain or service.

- Auto MIPs can be created for VLANs at the default maintenance domain level if they are not attached to a specific MA, or they can be:
  - Created at a specified level for a maintenance domain or MA on any bridge port.
  - When a lower MEP-only option is given, auto MIPs are created at a specified level only where a MEP is configured at the next lower level for a maintenance domain or MA.
  - When an auto MIP command is not issued at the domain level or the MA level, auto MIPs are not created for a maintenance domain or MA level.
  - When an auto MIP command is not issued at the domain level but is issued at the MA level, auto MIPs are created at the MA level.
- Can be created per MA, which means that a MIP in a MA can be lower level than a MEP in another MA.
- Auto MIP creation command can be issued at the maintenance domain (level), which will create MIPs for all S-VLANs enabled or allowed on a port.
- Internal to a domain, not at the boundary.
- CFM frames received from MEPs and other MIPs are cataloged and forwarded, using both the wire and the Bridge relay.
- When MIP filtering is enabled, all CFM frames at a lower level are stopped and dropped, independent of whether they originate from the wire or the Bridge relay.
- All CFM frames at a higher level are forwarded, independent of whether they arrive from the wire or from the Bridge relay.
- Passive points respond only when triggered by CFM traceroute and loopback messages.
- Bridge-Brain MAC addresses are used.

If the port on which a MIP is configured is blocked by Spanning-Tree Protocol, the MIP can receive CFM messages and catalog them but cannot send them toward the Bridge relay. The MIP can receive and respond to CFM messages from the wire.

A MIP has only one level associated with it. The level filtering option is supported.

The figure below illustrates MEPs and MIPs at the operator, service provider, and customer levels.



CFM Messages

CFM uses standard Ethernet frames. CFM frames are distinguishable by EtherType and for multicast messages by MAC address. CFM frames are sourced, terminated, processed, and relayed by bridges. Routers can support only limited CFM functions.

Bridges that cannot interpret CFM messages forward them as normal data frames. All CFM messages are confined to a maintenance domain and to an MA. Three types of messages are supported:

- Continuity Check
- Linktrace
- Loopback

Continuity Check Messages

CFM continuity check messages (CCMs) are multicast heartbeat messages exchanged periodically among MEPs. They allow MEPs to discover other MEPs within a domain and allow MIPs to discover MEPs. CCMs are confined to a domain.

CFM CCMs have the following characteristics:

- Transmitted at a periodic interval by MEPs. The interval can be one of the following configurable values. The default is 10 seconds.
    - 10 seconds
    - 1 minute
    - 10 minutes

Note Default and supported interval values are platform dependent.

- Cataloged by MIPs at the same maintenance level.
- Terminated by remote MEPs at the same maintenance level.
- Unidirectional and do not solicit a response.
- Indicate the status of the bridge port on which the MEP is configured.

Linktrace Messages

CFM linktrace messages (LTMs) are multicast frames that a MEP transmits, at the request of an administrator, to track the path (hop-by-hop) to a destination MEP. They are similar to Layer 3 traceroute messages. LTMs allow the transmitting node to discover vital connectivity data about the path and allow the discovery of all MIPs along the path that belong to the same maintenance domain. LTMs are intercepted by maintenance points along the path and processed, transmitted, or dropped. At each hop where there is a maintenance point at the same level, a linktrace message reply (LTR) is transmitted back to the originating MEP. For each visible MIP, linktrace messages indicate ingress action, relay action, and egress action.

Linktrace messages include the destination MAC address, VLAN, and maintenance domain and they have Time To Live (TTL) to limit propagation within the network. They can be generated on demand using the CLI. LTMs are multicast and LTRs are unicast.

Loopback Messages

CFM loopback messages (LBMs) are unicast frames that a MEP transmits, at the request of an administrator, to verify connectivity to a particular maintenance point. A reply to a loopback message (LBR) indicates whether a destination is reachable but does not allow hop-by-hop discovery of the path. A loopback message is similar in concept to an Internet Control Message Protocol (ICMP) Echo (ping) message.

Because LBMs are unicast, they are forwarded like normal data frames except with the maintenance level restriction. If the outgoing port is known in the bridge's forwarding database and allows CFM frames at the message's maintenance level to pass through, the frame is sent out on that port. If the outgoing port is unknown, the message is broadcast on all ports in that domain.

A CFM LBM can be generated on demand using the CLI. The source of a loopback message must be a MEP; the destination may be a MEP or a MIP. Both CFM LBMs and LBRs are unicast. CFM LBMs specify the destination MAC address or MPID, VLAN, and maintenance domain.

Cross-Check Function

The cross-check function is a timer-driven post-provisioning service verification between dynamically discovered MEPs (via CCMs) and expected MEPs (via configuration) for a service. The cross-check function verifies that all endpoints of a multipoint or point-to-point service are operational. The function supports notifications when the service is operational; otherwise it provides alarms and notifications for unexpected or missing endpoints.

The cross-check function is performed one time. You must initiate the cross-check function from the CLI every time you want a service verification.

SNMP Traps

The support provided by the Cisco IOS software implementation of CFM traps is Cisco proprietary information. MEPs generate two types of Simple Network Management Protocol (SNMP) traps, continuity check (CC) traps and cross-check traps.

CC Traps

- MEP up--Sent when a new MEP is discovered, the status of a remote port changes, or connectivity from a previously discovered MEP is restored after interruption.
- MEP down--Sent when a timeout or last gasp event occurs.
- Cross-connect--Sent when a service ID does not match the VLAN.
- Loop--Sent when a MEP receives its own CCMs.
- Configuration error--Sent when a MEP receives a continuity check with an overlapping MPID.

Cross-Check Traps

- Service up--Sent when all expected remote MEPs are up in time.
- MEP missing--Sent when an expected MEP is down.
- Unknown MEP--Sent when a CCM is received from an unexpected MEP.

Ethernet CFM and Ethernet OAM Interworking

- Ethernet Virtual Circuit, page 40
- OAM Manager, page 41

Ethernet Virtual Circuit

An EVC as defined by the Metro Ethernet Forum is a port-level point-to-point or multipoint-to-multipoint Layer 2 circuit. EVC status can be used by a CE device either to find an alternative path in to the service provider network or in some cases, to fall back to a backup path over Ethernet or over another alternative service such as Frame Relay or ATM.

OAM Manager

The OAM manager is an infrastructure element that streamlines interaction between OAM protocols. The OAM manager requires two interworking OAM protocols; for example, Ethernet CFM 802.1ag and link level Ethernet OAM 802.3ah. Interaction is unidirectional from the OAM manager to the CFM protocol and the only information exchanged is the user network interface (UNI) port status. Additional port status values available include

- REMOTE_EE--Remote excessive errors
- LOCAL_EE--Local excessive errors
- TEST--Either remote or local loopback

After CFM receives the port status, it communicates that status across the CFM domain.

HA Feature Support in CFM

In access and service provider networks using Ethernet technology, HA is a requirement, especially on Ethernet OAM components that manage EVC connectivity. End-to-end connectivity status information is critical and must be maintained on a hot standby route processor (RP).

Note A hot standby RP has the same software image as the active RP and supports synchronization of line card, protocol, and application state information between RPs for supported features and protocols.

End-to-end connectivity status is maintained on the CE, PE, and access aggregation PE (uPE) network nodes based on information received by protocols such as Ethernet LMI, CFM, and 802.3ah. This status information is used to either stop traffic or switch to backup paths when an EVC is down.

Every transaction involves either accessing or updating data among various databases. If the database is synchronized across active and standby modules, the modules are transparent to clients.

The Cisco IOS infrastructure provides various component application program interfaces (APIs) that help to maintain a hot standby RP. Metro Ethernet HA clients E-LMI HA/ISSU, CFM HA/ISSU, and 802.3ah HA/ISSU interact with these components, update the database, and trigger necessary events to other components.

Benefits of CFM HA

- Elimination of network downtime for Cisco IOS software image upgrades, allowing for faster upgrades that result in high availability.
- Elimination of resource scheduling challenges associated with planned outages and late night maintenance windows.
- Accelerated deployment of new services and applications and facilitation of faster implementation of new features, hardware, and fixes than if HA wasn’t supported.
- Reduced operating costs due to outages while delivering high service levels.
- CFM updates its databases and controls its own HA messaging and versioning, and this control facilitates maintenance.

- CFM HA in a Metro Ethernet Network, page 42
- NSF SSO Support in IEEE CFM, page 42
- ISSU Support in IEEE CFM, page 42

CFM HA in a Metro Ethernet Network

A standalone CFM implementation does not have explicit HA requirements. When CFM is implemented on a CE or PE with E-LMI, CFM must maintain the EVC state, which requires HA because the EVC state is critical in maintaining end-to-end connectivity. CFM configures the platform with maintenance level, domain, and maintenance point, learns the remote maintenance point information, and maps it to the appropriate EVC. CFM then aggregates data received from all remote ports and updates E-LMI; consequently HA requirements vary for CE and PE.

None of the protocols used in a Metro Ethernet Network (MEN) take action based on an EVC state, but a CE device that uses the E-LMI protocol and receives EVC information will stop sending traffic to the MEN when the EVC is down. When an EVC is down, the CE may also use a backup network, if available.

The CE receives the EVC ID, associated customer VLANs, UNI information, EVC state, and remote UNI ID and state from the MEN. The CE relies on the EVC state to send or stop traffic to the MEN via E-LMI.

The PE has EVC configuration and associated customer VLAN information and derives the EVC state and remote UNI from CFM. This information is sent to the CE using E-LMI.

Note PEs and CEs running 802.3ah OAM must maintain the port state so peers are not affected by a switchover. This information is also sent to remote nodes in CFM CCMs.

NSF SSO Support in IEEE CFM

The redundancy configurations SSO and NSF are both supported in IEEE CFM and are automatically enabled. A switchover from an active to a standby RP occurs when the active RP fails, is removed from the networking device, or is manually taken down for maintenance. NSF interoperates with the SSO feature to minimize network downtime following a switchover. The primary function of Cisco NSF is to continue forwarding packets following an RP switchover.

For detailed information about SSO, see the “Stateful Switchover” chapter of the *Cisco IOS High Availability Configuration Guide*. For detailed information about the NSF feature, see the “Cisco Nonstop Forwarding” chapter of the *Cisco IOS High Availability Configuration Guide*.

ISSU Support in IEEE CFM

ISSU allows you to perform a Cisco IOS software upgrade or downgrade without disrupting packet flow. CFM performs a bulk update and a runtime update of the continuity check database to the standby RP, including adding, deleting, or updating a row. This checkpoint data requires ISSU capability to transform messages from one release to another. All the components that perform active RP to standby RP updates using messages require ISSU support.

ISSU is automatically enabled in CFM and lowers the impact that planned maintenance activities have on network availability by allowing software changes while the system is in service. For detailed information about ISSU, see the “Cisco IOS In Service Software Upgrade Process” chapter of the *Cisco IOS High Availability Configuration Guide*.

IEEE CFM Bridge Domain Support

Note When an EFP with an inward-facing MEP (a PE interface toward a uPE interface) is configured with the default EFP encapsulation, the inward-facing MEPs on both ends receive CCMs from each other at a preset time interval. However, with the default encapsulation configured, packets are dropped and as a result, the CCMs are dropped at the ingress port. To stop packets from being dropped, at the default EFP configure the desired encapsulation using the cfm encapsulation command.

An Ethernet flow point (EFP) or a service instance is a logical demarcation point of a bridge domain on an interface. VLAN tags are used to match and map traffic to the EFP. VLAN IDs have local significance per port similar to ATM/Frame Relay virtual circuits. CFM is supported on a bridge domain associated with an EFP. The association between the bridge domain and the EFP allows CFM to use the encapsulation on the EFP. All EFPs in the same bridge domain form a broadcast domain. The bridge domain ID determines the broadcast domain.

The distinction between a VLAN port and the EFP is the encapsulation. VLAN ports use a default dot1q encapsulation. For EFPs untagged, single tagged, and double tagged, encapsulation exists with dot1q and IEEE dot1ad EtherTypes. Different EFPs belonging to the same bridge domain can use different encapsulations.

Note IEEE CFM support for bridge domains is available only on ES20 and ES40 line cards.

Untagged CFM packets can be associated with a maintenance point. An incoming untagged customer CFM packet has an EtherType of CFM and is mapped to an EVC (bridge domain) based on the encapsulation configured on the EFP. The EFP can be configured specifically to recognize these untagged packets.

Switchport VLANs and EFPs configured with bridge domains handle MEPs and MIPs for a service independently. The bridge domain-to-VLAN space mapping is different for different platforms. For bridge domain and switchport VLAN interworking (maintenance points, ingress and egress are on both switchports and EFPs), a bridge domain-VLAN service should be configured on platforms where the bridge domain and switchport VLAN represent the same broadcast domain. On the Cisco 7600 series router, a bridge domain and a switchport VLAN with the same number form a single broadcast domain.

How to Set Up IEEE Ethernet CFM in a Service Provider Network

- Designing CFM Domains, page 44
- Configuring IEEE Ethernet CFM, page 46
- Configuring Ethernet OAM 802.3ah Interaction with CFM, page 125
- Configuring CFM for Bridge Domains, page 130

Designing CFM Domains

Note To have an operator, service provider, or customer domain is optional. A network may have a single domain or multiple domains. The steps listed here show the sequence when all three types of domains will be assigned.

- Knowledge and understanding of the network topology.
- Understanding of organizational entities involved in managing the network; for example, operators, service providers, network operations centers (NOCs), and customer service centers.
- Understanding of the type and scale of services to be offered.
- Agreement by all organizational entities on the responsibilities, roles, and restrictions for each organizational entity.
- Determination of the number of maintenance domains in the network.
- Determination of the nesting and disjoint maintenance domains.
- Assignment of maintenance levels and names to domains based on agreement between the service provider and operator or operators.
- Determination of whether the domain should be inward or outward.

SUMMARY STEPS

1. Determine operator level MIPs.
2. Determine operator level MEPs.
3. Determine service provider MIPs.
4. Determine service provider MEPs.
5. Determine customer MIPs.
6. Determine customer MEPs.

DETAILED STEPS

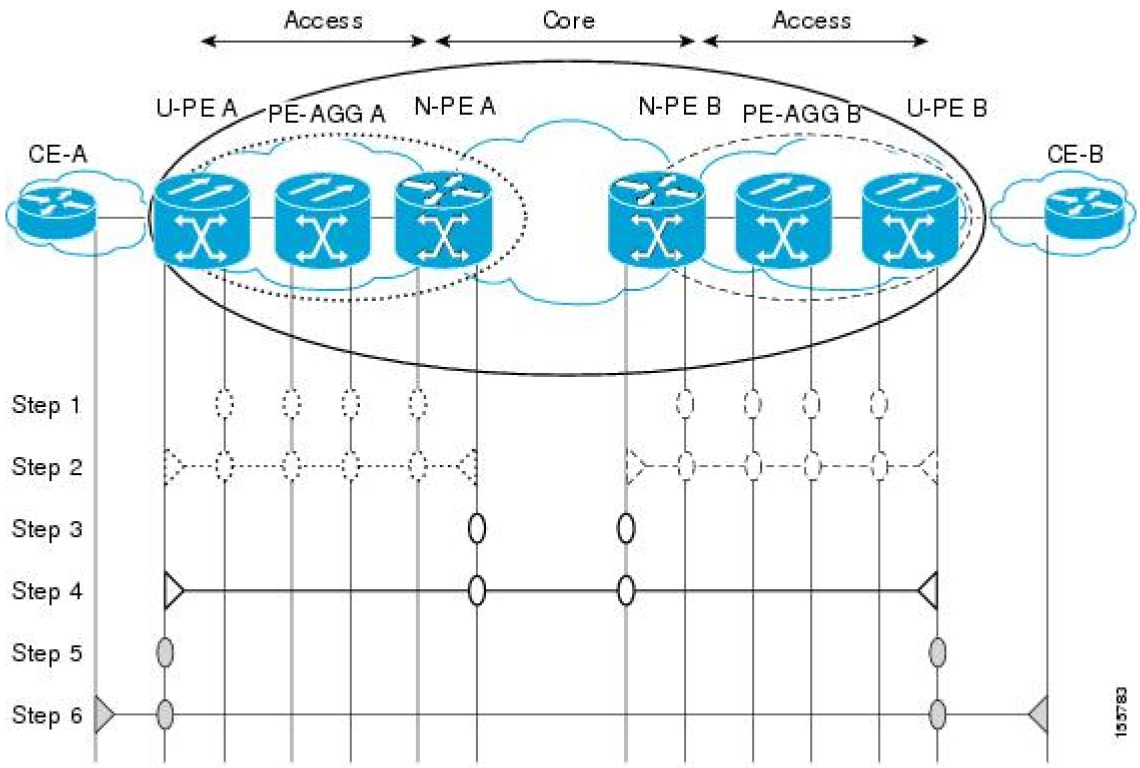
	Command or Action	Purpose
Step 1	Determine operator level MIPs.	Follow these steps: • Starting at lowest operator level domain, assign a MIP at every interface internal to the operator network to be visible to CFM. • Proceed to next higher operator level and assign MIPs. • Verify that every port that has a MIP at a lower level does not have maintenance points at a higher level. • Repeat steps a through d until all operator MIPs are determined.

	Command or Action	Purpose
Step 2	Determine operator level MEPs.	Follow these steps: • Starting at the lowest operator level domain, assign a MEP at every UNI that is part of a service instance. • Assign a MEP at the network to network interface (NNI) between operators, if there is more than one operator. • Proceed to next higher operator level and assign MEPs. • A port with a MIP at a lower level cannot have maintenance points at a higher level. A port with a MEP at a lower level should have either a MIP or MEP at a higher level.
Step 3	Determine service provider MIPs.	Follow these steps: • Starting at the lowest service provider level domain, assign service provider MIPs at the NNI between operators (if more than one). • Proceed to next higher service provider level and assign MIPs. • A port with a MIP at a lower level cannot have maintenance points at a higher level. A port with a MEP at a lower level should not have either a MIP or a MEP at a higher level.
Step 4	Determine service provider MEPs.	Follow these steps: • Starting at the lowest service provider level domain, assign a MEP at every UNI that is part of a service instance. • Proceed to next higher service provider level and assign MEPs. • A port with a MIP at a lower level cannot have maintenance points at a higher level. A port with a MEP at a lower level should have either a MIP or a MEP at a higher level.
Step 5	Determine customer MIPs.	Customer MIPs are allowed only on the UNIs at the uPEs if the service provider allows the customer to run CFM. Otherwise, the service provider can configure Cisco IOS devices to block CFM frames. • Configure a MIP on every uPE, at the UNI port, in the customer maintenance domain. • Ensure the MIPs are at a maintenance level that is at least one higher than the highest level service provider domain.
Step 6	Determine customer MEPs.	Customer MEPs are on customer equipment. Assign an outward facing MEP within an outward domain at the appropriate customer level at the handoff between the service provider and the customer.

- Examples, page 45
- Examples, page 45
- What to Do Next, page 163

Examples

The figure below shows an example of a network with a service provider and two operators, A and B. Three domains are to be established to map to each operator and the service provider. In this example, for simplicity we assume that the network uses Ethernet transport end to end. CFM, however, can be used with other transports.



Configuring IEEE Ethernet CFM

Provisioning the Network

- Provisioning the Network on the PE-AGG B, page 182
- Provisioning the Network on the N-PE B, page 184

Provisioning the Network for CE-A

Perform this task to prepare the network for Ethernet CFM.

To configure MIPs at different interfaces and service instances, you must configure an auto MIP under the domain and service.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **exit**
6. **ethernet cfm global**
7. **ethernet cfm ieee**
8. **ethernet cfm traceroute cache**
9. **ethernet cfm traceroute cache size** *entries*
10. **ethernet cfm traceroute cache hold-time** *minutes*
11. **snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**]
12. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**]
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and places the CLI in Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

Step 5	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 6	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 7	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued
Step 8	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 9	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.

	Command or Action	Purpose
Step 10	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 11	**snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM continuity check events.
Step 12	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown`	Enables SNMP trap generation for Ethernet CFM continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.

Step 13	**end**
	
	Example:
	
	`Router(config)# end`
	
	Example:
	
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning the Network for U-PE A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **exit**
5. **ethernet cfm domain** *domain-name* **level** *level-id*
6. **mep archive-hold-time** *minutes*
7. **exit**
8. **ethernet cfm mip** {**auto-create level** *level-id* **vlan** {*vlan-id*| *vlan-id-vlan-id*| **,** *vlan-id-vlan-id*}[**lower-mep-only**] [**sender-id chassis**]| **filter**}
9. **ethernet cfm domain** *domain-name* **level** *level-id*
10. **mep archive-hold-time** *minutes*
11. **mip auto-create** [**lower-mep-only**]
12. **exit**
13. **ethernet cfm global**
14. **ethernet cfm ieee**
15. **ethernet cfm traceroute cache**
16. **ethernet cfm traceroute cache size** *entries*
17. **ethernet cfm traceroute cache hold-time** *minutes*
18. **interface** *type number*
19. **ethernet cfm mip level** *level-id*
20. **exit**
21. **snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**]
22. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**]
23. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and places the CLI in Ethernet CFM configuration mode.

Step 4	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 5	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain at a particular maintenance level and places the CLI in Ethernet CFM configuration mode.
Step 6	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

Step 7	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 8	**ethernet cfm mip** {**auto-create level** *level-id* **vlan** {*vlan-id*| *vlan-id-vlan-id*| **,** *vlan-id-vlan-id*}[**lower-mep-only**] [**sender-id chassis**]| **filter**} **Example:** `Router(config)# ethernet cfm mip auto-create level 1 vlan 2000`	Dynamically creates a MIP and provisions it globally at a specified maintenance level for VLAN IDs that are not associated with specific MAs or enables level filtering.
Step 9	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorA level 1`	Defines a CFM maintenance domain at a particular maintenance level and places the CLI in Ethernet CFM configuration mode.
Step 10	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 11	**mip auto-create** [**lower-mep-only**] **Example:** `Router(config-ecfm)# mip auto-create`	Enables the dynamic creation of a MIP at a maintenance domain level.

Step 12	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 13	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 14	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued
Step 15	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 16	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 17	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 18	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet4/2`	Specifies an interface and places the CLI in interface configuration mode.
Step 19	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.

	Command or Action	Purpose

Step 20	**exit**
	
	Example:
	`Router(config-if)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 21	**snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.
Step 22	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown`	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.

Step 23	**end**
	
	Example:
	`Router(config)# end`
	
	Example:
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning the Network for PE-AGG A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mip auto-create** [**lower-mep-only**]
5. **mep archive-hold-time** *minutes*
6. **exit**
7. **ethernet cfm global**
8. **ethernet cfm ieee**
9. **interface** *type number*
10. **ethernet cfm mip level** *level-id*
11. **interface** *type number*
12. **ethernet cfm mip level** *level-id*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorA level 1`	Defines a domain and places the CLI in Ethernet CFM configuration mode.
Step 4	**mip auto-create** [**lower-mep-only**] **Example:** `Router(config-ecfm)# mip auto-create`	Enables the dynamic creation of a MIP at a maintenance domain level.

	Command or Action	Purpose
Step 5	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

Step 6	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 7	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 8	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued
Step 9	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet3/1`	Specifies an interface and places the CLI in interface configuration mode.
Step 10	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.
Step 11	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet4/1`	Specifies an interface.

	Command or Action	Purpose
Step 12	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.

Step 13	**end**
	
	Example:
	`Router(config-if)# end`
	
	Example:
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning the Network for N-PE A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **mip auto-create** [**lower-mep-only**]
6. **exit**
7. **ethernet cfm domain** *domain-name* **level** *level-id*
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
11. **ethernet cfm ieee**
12. **ethernet cfm traceroute cache**
13. **ethernet cfm traceroute cache size** *entries*
14. **ethernet cfm traceroute cache hold-time** *minutes*
15. **interface** *type number*
16. **ethernet cfm mip level** *level-id*
17. **exit**
18. **snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**]
19. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**]
20. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain and level and places the CLI in Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**mip auto-create** [**lower-mep-only**] **Example:** `Router(config-ecfm)# mip auto-create`	Enables the dynamic creation of a MIP at a maintenance domain level.

Step 6	**exit**
	
	Example:
	`Router(config-ecfm)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 7	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorA level 1`	Defines a CFM maintenance domain and level and places the CLI in Ethernet CFM configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

Step 9	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 10	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 11	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued
Step 12	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.

	Command or Action	Purpose
Step 13	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 14	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 15	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet3/0`	Specifies an interface and places the CLI in interface configuration mode.
Step 16	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.

Step 17	**exit**
	
	Example:
	
	`Router(config-if)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 18	**snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.

	Command or Action	Purpose
Step 19	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown`	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.

Step 20	**end**
	
	Example:
	
	`Router(config)# end`
	
	Example:
	
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning the Network for U-PE B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **exit**
5. **ethernet cfm domain** *domain-name* **level** *level-id*
6. **mep archive-hold-time** *minutes*
7. **exit**
8. **ethernet cfm domain** *domain-name* **level** *level-id*
9. **mep archive-hold-time** *minutes*
10. **exit**
11. **ethernet cfm global**
12. **ethernet cfm ieee**
13. **ethernet cfm traceroute cache**
14. **ethernet cfm traceroute cache size** *entries*
15. **ethernet cfm traceroute cache hold-time** *minutes*
16. **interface** *type number*
17. **ethernet cfm mip level** *level-id*
18. **exit**
19. **snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**]
20. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**]
21. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.

Step 4	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 5	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 6	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

Step 7	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 8	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorB level 2`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 9	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

Step 10	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 11	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 12	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued
Step 13	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.

	Command or Action	Purpose
Step 14	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 15	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 16	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet2/0`	Specifies an interface and places the CLI in interface configuration mode.
Step 17	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a manual MIP.

Step 18	**exit**
	
	Example:
	
	`Router(config-if)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 19	**snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.

	Command or Action	Purpose
Step 20	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown`	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.

Step 21	**end**
	
	Example:
	
	`Router(config)# end`
	
	Example:
	
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning the Network for PE-AGG B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **mip auto-create** [**lower-mep-only**]
6. **exit**
7. **ethernet cfm global**
8. **ethernet cfm ieee**
9. **interface** *type number*
10. **ethernet cfm mip level** *level-id*
11. **interface** *type number*
12. **ethernet cfm mip level** *level-id*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorB level 2`	Defines a domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**mip auto-create** [**lower-mep-only**] **Example:** `Router(config-ecfm)# mip auto-create`	Enables the dynamic creation of a MIP at a maintenance domain level.

Step 6	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 7	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 8	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued
Step 9	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet1/1`	Specifies an interface and places the CLI in interface configuration mode.
Step 10	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a manual MIP.
Step 11	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet2/1`	Specifies an interface.
Step 12	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.

Step 13	**end**
	
	Example:
	
	`Router(config-if)# end`
	
	Example:
	
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning the Network for U-PE B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **exit**
6. **ethernet cfm domain** *domain-name* **level** *level-id*
7. **mep archive-hold-time** *minutes*
8. **mip auto-create** [**lower-mep-only**]
9. **exit**
10. **ethernet cfm global**
11. **ethernet cfm ieee**
12. **ethernet cfm traceroute cache**
13. **ethernet cfm traceroute cache size** *entries*
14. **ethernet cfm traceroute cache hold-time** *minutes*
15. **interface** *type number*
16. **ethernet cfm mip level** *level-id*
17. **exit**
18. **snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**]
19. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**]
20. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

Step 5	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 6	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorB level 2`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 7	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 8	**mip auto-create** [**lower-mep-only**] **Example:** `Router(config-ecfm)# mip auto-create`	Enables the dynamic creation of a MIP at a maintenance domain level.

	Command or Action	Purpose

Step 9	**exit**
	
	Example:
	`Router(config-ecfm)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 10	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 11	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued
Step 12	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 13	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 14	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.

	Command or Action	Purpose
Step 15	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet1/2`	Specifies an interface and places the CLI in interface configuration mode.
Step 16	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.

Step 17	**exit**
	
	Example:
	
	`Router(config-if)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 18	**snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**][**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.
Step 19	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown`	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.

	Command or Action	Purpose

Step 20	**end**
	
	Example:
	`Router(config)# end`
	
	Example:
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning the Network for CE-B

SUMMARY STEPS

1.
2. **enable**
3. **configure terminal**
4. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
5. **mep archive-hold-time** *minutes*
6. **exit**
7. **ethernet cfm global**
8. **ethernet cfm ieee**
9. **ethernet cfm traceroute cache**
10. **ethernet cfm traceroute cache size** *entries*
11. **ethernet cfm traceroute cache hold-time** *minutes*
12. **snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**]
13. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**]
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1		**CE-B**
Step 2	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 3	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 4	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain Customer level 7 direction outward`	Defines an outward CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 5	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

Step 6	**exit**
	
	Example:
	`Router(config-ecfm)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 7	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 8	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued

	Command or Action	Purpose
Step 9	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 10	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 11	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 12	**snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**][**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.
Step 13	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown`	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.

Step 14	**end**
	
	Example:
	
	`Router(config)# end`
	
	Example:
	
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning Service

Provisioning Service for CE-A

Perform this task to set up service for Ethernet CFM. Optionally, when this task is completed, you may configure and enable the cross-check function. To perform this optional task, see "Configuring and Enabling the Cross-Check Function".

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
5. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
6. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
7. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
8. **exit**
9. **mep archive-hold-time** *minutes*
10. **exit**
11. **ethernet cfm global**
12. **ethernet cfm ieee**
13. **ethernet cfm traceroute cache**
14. **ethernet cfm traceroute cache size** *entries*
15. **ethernet cfm traceroute cache hold-time** *minutes*
16. **interface** *type number*
17. **ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*}
18. Do one of the following:
    - **switchport**
    - **switchport mode trunk**
19. **ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*}
20. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a specified maintenance level and places the CLI in Ethernet CFM configuration mode.
Step 4	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Router(config-ecfm)# service Customer1 vlan 101 direction down`	Configures a maintenance association within a maintenance domain and places the CLI into CFM service configuration mode.
Step 5	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check`	Enables the transmission of CCMs.
Step 6	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check interval 10s`	Configures the time period between CCM transmissions.
Step 7	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check loss-threshold 10`	Sets the number of CCMs that should be missed before declaring that a remote MEP is down.

Step 8	**exit**
	Example:
	`Router(config-ecfm-srv)# exit`
	Example:
	`Router(config-ecfm)#`
		Returns the CLI to Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 9	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

Step 10	**exit**
	Example:
	`Router(config-ecfm)# exit`
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 11	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 12	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued
Step 13	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 14	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.

	Command or Action	Purpose
Step 15	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 16	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/3`	Specifies an interface and places the CLI in interface configuration mode.
Step 17	**ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep domain Customer mpid 701 vlan 100`	Sets a port as internal to a maintenance domain and defines it as a MEP.

Step 18	Do one of the following:
	• **switchport**
	• **switchport mode trunk**
	
	Example:
	
	`Router(config-if)# switchport`
	
	Example:
	
	`Router(config-if)# switchport mode trunk`
		Specifies a switchport or alternatively, specifies a trunking VLAN Layer 2 interface.

Step 19	**ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep domain Customer mpid 701 vlan 100`	Sets a port as internal to a maintenance domain and defines it as a MEP.

	Command or Action	Purpose

Step 20	**end**
	
	Example:
	`Router(config-if)# end`
	
	Example:
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning Service for U-PE A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **exit**
5. **exit**
6. **ethernet cfm domain** *domain-name* **level** *level-id*
7. **mep archive-hold-time** *minutes*
8. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
9. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
10. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
11. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
12. **exit**
13. **exit**
14. **ethernet cfm domain** *domain-name* **level** *level-id*
15. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
16. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
17. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
18. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
19. **exit**
20. **mep archive-hold-time** *minutes*
21. **exit**
22. **ethernet cfm global**
23. **ethernet cfm ieee**
24. **ethernet cfm traceroute cache**
25. **ethernet cfm traceroute cache size** *entries*
26. **ethernet cfm traceroute cache hold-time** *minutes*
27. **interface** *type number*
28. **ethernet cfm mip level** *level-id*
29. **ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*}
30. **interface** *type number*
31. **ethernet cfm mip level** *level-id*
32. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.

Step 4	**exit**
	
	Example:
	`Router(config-ecfm-srv)# exit`
	
	Example:
	`Router(config-ecfm)#`
		Returns the CLI to Ethernet CFM configuration mode.

Step 5	**exit**
	
	Example:
	`Router(config-ecfm)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 6	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 7	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 8	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Router(config-ecfm)# service MetroCustomer1 vlan 101`	Configures a maintenance association within a maintenance domain and places the CLI into CFM service configuration mode.
Step 9	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check`	Enables the transmission of CCMs.
Step 10	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check interval 10s`	Configures the time period between CCM transmissions.
Step 11	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check loss-threshold 10`	Sets the number of CCMs that should be missed before declaring that a remote MEP is down.

	Command or Action	Purpose

Step 12	**exit**
	
	Example:
	`Router(config-ecfm-srv)# exit`
	
	Example:
	`Router(config-ecfm)#`
		Returns the CLI to Ethernet CFM configuration mode.

Step 13	**exit**
	
	Example:
	`Router(config-ecfm)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 14	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorA level 1`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 15	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Router(config-ecfm)# service MetroCustomer1OpA vlan 101`	Configures a maintenance association within a maintenance domain and places the CLI into CFM service configuration mode.
Step 16	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check`	Enables the transmission of CCMs.

	Command or Action	Purpose
Step 17	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check interval 10s`	Configures the time period between CCM transmissions.
Step 18	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check loss-threshold 10`	Sets the number of CCMs that should be missed before declaring that a remote MEP is down.

Step 19	**exit**
	
	Example:
	
	`Router(config-ecfm-srv)# exit`
	
	Example:
	
	`Router(config-ecfm)#`
		Returns the CLI to Ethernet CFM configuration mode.

Step 20	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

Step 21	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 22	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 23	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued
Step 24	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 25	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 26	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 27	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet3/2`	Specifies an interface and places the CLI in interface configuration mode.
Step 28	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 7`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.

	Command or Action	Purpose
Step 29	**ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep domain Customer mpid 701 vlan 100`	Sets a port as internal to a maintenance domain and defines it as a MEP.
Step 30	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet 4/2`	Specifies an interface.
Step 31	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.

Step 32	**end**
	
	Example:
	
	`Router(config-if)# end`
	
	Example:
	
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning Service for PE-AGG A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **mip auto-create** [**lower-mep-only**]
6. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
7. **exit**
8. **exit**
9. **ethernet cfm global**
10. **ethernet cfm ieee**
11. **interface** *type number*
12. **ethernet cfm mip level** *level-id*
13. **interface** *type number*
14. **ethernet cfm mip level** *level-id*
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorA level 1`	Defines a domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

	Command or Action	Purpose
Step 5	**mip auto-create** [**lower-mep-only**] **Example:** `Router(config-ecfm)# mip auto-create`	Enables the dynamic creation of a MIP at a maintenance domain level.
Step 6	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Router(config-ecfm)# service MetroCustomer1OpA vlan 101`	Configures a maintenance association within a maintenance domain and places the CLI into CFM service configuration mode.

Step 7	**exit**
	
	Example:
	
	`Router(config-ecfm-srv)# exit`
	
	Example:
	
	`Router(config-ecfm)#`
		Returns the CLI to Ethernet CFM configuration mode.

Step 8	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 9	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 10	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued

	Command or Action	Purpose
Step 11	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet3/1`	Specifies an interface and places the CLI in interface configuration mode.
Step 12	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.
Step 13	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet4/1`	Specifies an interface.
Step 14	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.

Step 15	**end**
	
	Example:
	
	`Router(config-if)# end`
	
	Example:
	
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning Service for N-PE A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **mip auto-create** [**lower-mep-only**]
6. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
7. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
8. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
9. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
10. **exit**
11. **exit**
12. **ethernet cfm domain** *domain-name* **level** *level-id*
13. **mep archive-hold-time** *minutes*
14. **mip auto-create** [**lower-mep-only**]
15. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
16. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
17. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
18. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
19. **exit**
20. **exit**
21. **ethernet cfm global**
22. **ethernet cfm ieee**
23. **ethernet cfm traceroute cache**
24. **ethernet cfm traceroute cache size** *entries*
25. **ethernet cfm traceroute cache hold-time** *minutes*
26. **interface** *type number*
27. **ethernet cfm mip level** *level-id*
28. **interface** *type number*
29. **ethernet cfm mip level** *level-id*
30. **ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*}
31. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**mip auto-create** [**lower-mep-only**] **Example:** `Router(config-ecfm)# mip auto-create`	Enables the dynamic creation of a MIP at a maintenance domain level.
Step 6	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Router(config-ecfm)# service MetroCustomer1 vlan 101`	Configures a maintenance association within a maintenance domain and places the CLI into CFM service configuration mode.
Step 7	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check`	Enables the transmission of CCMs.

	Command or Action	Purpose
Step 8	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check interval 10s`	Configures the time period between CCM transmissions.
Step 9	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check loss-threshold 10`	Sets the number of CCMs that should be missed before declaring that a remote MEP is down.

Step 10	**exit**
	
	Example:
	`Router(config-ecfm-srv)# exit`
	
	Example:
	`Router(config-ecfm)#`
		Returns the CLI to Ethernet CFM configuration mode.

Step 11	**exit**
	
	Example:
	`Router(config-ecfm)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 12	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorA level 1`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 13	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 14	**mip auto-create** [**lower-mep-only**] **Example:** `Router(config-ecfm)# mip auto-create`	Enables the dynamic creation of a MIP at a maintenance domain level.
Step 15	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Router(config-ecfm)# service MetroCustomer1OpA vlan 101`	Configures a maintenance association within a maintenance domain and places the CLI into CFM service configuration mode.
Step 16	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check`	Enables the transmission of CCMs.
Step 17	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check interval 10s`	Configures the time period between CCM transmissions.
Step 18	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check loss-threshold 10`	Sets the number of CCMs that should be missed before declaring that a remote MEP is down.

	Command or Action	Purpose

Step 19	**exit**
	
	Example:
	
	`Router(config-ecfm-srv)# exit`
	
	Example:
	
	`Router(config-ecfm)#`
		Returns the CLI to Ethernet CFM configuration mode.

Step 20	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 21	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 22	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued
Step 23	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 24	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.

	Command or Action	Purpose
Step 25	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 26	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet3/0`	Specifies an interface and places the CLI in interface configuration mode.
Step 27	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a manual MIP. • This is an optional manual MIP
Step 28	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet4/0`	Specifies an interface.
Step 29	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 4`	Provisions a manual MIP. • This is an optional manual MIP
Step 30	**ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep domain Customer mpid 701 vlan 100`	Sets a port as internal to a maintenance domain and defines it as a MEP.

	Command or Action	Purpose

Step 31	**end**
	
	Example:
	
	`Router(config-if)# end`
	
	Example:
	
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning Service for U-PE B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **exit**
5. **ethernet cfm domain** *domain-name* **level** *level-id*
6. **mep archive-hold-time** *minutes*
7. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
8. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
9. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
10. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
11. **exit**
12. **exit**
13. **ethernet cfm domain** *domain-name* **level** *level-id*
14. **mep archive-hold-time** *minutes*
15. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
16. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
17. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
18. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
19. **exit**
20. **exit**
21. **ethernet cfm global**
22. **ethernet cfm ieee**
23. **ethernet cfm traceroute cache**
24. **ethernet cfm traceroute cache size** *entries*
25. **ethernet cfm traceroute cache hold-time** *minutes*
26. **interface** *type number*
27. **ethernet cfm mip level** *level-id*
28. **ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*}
29. **interface** *type number*
30. **ethernet cfm mip level** *level-id*
31. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.

Step 4	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 5	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 6	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

	Command or Action	Purpose
Step 7	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Router(config-ecfm)# service Customer1 vlan 101 direction down`	Configures a maintenance association within a maintenance domain and places the CLI into CFM service configuration mode.
Step 8	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check`	Enables the transmission of CCMs.
Step 9	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check interval 10s`	Configures the time period between CCM transmissions.
Step 10	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check loss-threshold 10`	Sets the number of CCMs that should be missed before declaring that a remote MEP is down.

Step 11	**exit**
	
	Example:
	
	`Router(config-ecfm-srv)# exit`
	
	Example:
	
	`Router(config-ecfm)#`
		Returns the CLI to Ethernet CFM configuration mode.

	Command or Action	Purpose

Step 12	**exit**
	
	Example:
	`Router(config-ecfm)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 13	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorB level 2`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 14	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 15	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Router(config-ecfm)# service MetroCustomer1 vlan 101`	Configures a maintenance association within a maintenance domain and places the CLI into CFM service configuration mode.
Step 16	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check`	Enables the transmission of CCMs.
Step 17	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check interval 10s`	Configures the time period between CCM transmissions.

	Command or Action	Purpose
Step 18	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check loss-threshold 10`	Sets the number of CCMs that should be missed before declaring that a remote MEP is down.

Step 19	**exit**
	
	Example:
	
	`Router(config-ecfm-srv)# exit`
	
	Example:
	
	`Router(config-ecfm)#`
		Returns the CLI to Ethernet CFM configuration mode.

Step 20	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 21	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 22	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued
Step 23	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.

	Command or Action	Purpose
Step 24	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 25	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 26	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet1/0`	Specifies an interface and places the CLI in interface configuration mode.
Step 27	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 7`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.
Step 28	**ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep domain Customer mpid 701 vlan 100`	Sets a port as internal to a maintenance domain and defines it as a MEP.
Step 29	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet2/0`	Specifies an interface.
Step 30	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.

	Command or Action	Purpose

Step 31	**end**
	
	Example:
	`Router(config)# end`
	
	Example:
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning Service for PE-AGG B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
6. **exit**
7. **exit**
8. **ethernet cfm global**
9. **ethernet cfm ieee**
10. **interface** *type number*
11. **ethernet cfm mip level** *level-id*
12. **interface** *type number*
13. **ethernet cfm mip level** *level-id*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorB level 2`	Defines a domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 65`	Set the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Router(config-ecfm)# service MetroCustomer1 vlan 101`	Configures a maintenance association within a maintenance domain and places the CLI into CFM service configuration mode.

Step 6	**exit**
	
	Example:
	`Router(config-ecfm-srv)# exit`
	
	Example:
	`Router(config-ecfm)#`
		Returns the CLI to Ethernet CFM configuration mode.

	Command or Action	Purpose

Step 7	**exit**
	
	Example:
	`Router(config-ecfm)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 8	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 9	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued
Step 10	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet1/1`	Specifies an interface and places the CLI in interface configuration mode.
Step 11	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.
Step 12	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet2/1`	Specifies an interface.
Step 13	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.

	Command or Action	Purpose

Step 14	**end**
	
	Example:
	`Router(config-if)# end`
	
	Example:
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning Service for N-PE B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
6. **exit**
7. **ethernet cfm domain** *domain-name* **level** *level-id*
8. **mep archive-hold-time** *minutes*
9. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
10. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
11. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
12. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
13. **exit**
14. **exit**
15. **ethernet cfm global**
16. **ethernet cfm ieee**
17. **ethernet cfm traceroute cache**
18. **ethernet cfm traceroute cache size** *entries*
19. **ethernet cfm traceroute cache hold-time** *minutes*
20. **interface** *type number*
21. **ethernet cfm mip level** *level-id*
22. **interface** *type number*
23. **ethernet cfm mip level** *level-id*
24. **ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*}
25. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Router(config-ecfm)# service MetroCustomer1 vlan 101`	Configures a maintenance association within a maintenance domain and places the CLI into CFM service configuration mode.

Step 6	**exit**
	
	Example:
	
	`Router(config-ecfm-srv)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 7	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorB level 2`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Router(config-ecfm)# service MetroCustomer1OpB vlan 101`	Configures a maintenance association within a maintenance domain and places the CLI into CFM service configuration mode.
Step 10	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check`	Enables the transmission of CCMs.
Step 11	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check interval 10s`	Configures the time period between CCM transmissions.
Step 12	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check loss-threshold 10`	Sets the number of CCMs that should be missed before declaring that a remote MEP is down.

	Command or Action	Purpose

Step 13	**exit**
	
	Example:
	`Router(config-ecfm-srv)# exit`
	
	Example:
	`Router(config-ecfm)#`
		Returns the CLI to Ethernet CFM configuration mode.

Step 14	**exit**
	
	Example:
	`Router(config-ecfm)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 15	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 16	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued
Step 17	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 18	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.

	Command or Action	Purpose
Step 19	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 20	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet1/2`	Specifies an interface and places the CLI in interface configuration mode.
Step 21	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.
Step 22	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet2/2`	Specifies an interface.
Step 23	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 4`	Provisions a manual MIP. • This is an optional use of a manual MIP and can override auto MIP configuration.
Step 24	**ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep domain Customer mpid 701 vlan 100`	Sets a port as internal to a maintenance domain and defines it as a MEP.

	Command or Action	Purpose

Step 25	**end**
	
	Example:
	`Router(config-if)#`
	
	Example:
	`Router#`
		Returns the CLI to privileged EXEC mode.

Provisioning Service for CE-B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
4. **mep archive-hold-time** *minutes*
5. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
6. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
7. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
8. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
9. **exit**
10. **exit**
11. **ethernet cfm global**
12. **ethernet cfm ieee**
13. **ethernet cfm traceroute cache**
14. **ethernet cfm traceroute cache size** *entries*
15. **ethernet cfm traceroute cache hold-time** *minutes*
16. **interface** *type number*
17. **ethernet cfm mep level** *level-id* [**inward**| **outward domain** *domain-name*] **mpid** *id* **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
18. Do one of the following:
    - **switchport**
    -
    - **switchport mode trunk**
19. **ethernet cfm mep level** *level-id* [**inward**| **outward domain** *domain-name*] **mpid** *id* **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
20. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain Customer level 7 direction outward`	Defines a CFM maintenance domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ecfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Router(config-ecfm)# service Customer1 vlan 101 direction down`	Configures a maintenance association within a maintenance domain and places the CLI into CFM service configuration mode.
Step 6	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check`	Enables the transmission of CCMs.

	Command or Action	Purpose
Step 7	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check interval 10s`	Configures the time period between CCM transmissions.
Step 8	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check loss-threshold 10`	Sets the number of CCMs that should be missed before declaring that a remote MEP is down.

Step 9	**exit**
	
	Example:
	`Router(config-ecfm-srv)# exit`
	
	Example:
	`Router(config-ecfm)#`
		Returns the CLI to Ethernet CFM configuration mode.

Step 10	**exit**
	
	Example:
	`Router(config-ecfm)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 11	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 12	**ethernet cfm ieee** **Example:** `Router(config)# ethernet cfm ieee`	Enables the CFM IEEE version of CFM. • This command is automatically issued when the **ethernet cfm global** command is issued
Step 13	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 14	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 15	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 16	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/1`	Specifies an interface and places the CLI in interface configuration mode.
Step 17	**ethernet cfm mep level** *level-id* [**inward**| **outward domain** *domain-name*] **mpid** *id* **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep level 7 outward domain Customer mpid 701 vlan 100`	Sets an interface as a domain boundary.

	Command or Action	Purpose

Step 18	Do one of the following:
	• **switchport**
	•
	• **switchport mode trunk**
	
	Example:
	`Router(config-if)# switchport`
	
	Example:
	
	Example:
	`Router(config-if)# switchport mode trunk`
		Specifies a switchport or alternatively, specifies a trunking VLAN Layer 2 interface.

Step 19	**ethernet cfm mep level** *level-id* [**inward**| **outward domain** *domain-name*] **mpid** *id* **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep level 7 outward domain Customer mpid 701 vlan 100`	Provisions an interface as a domain boundary.

Step 20	**end**
	
	Example:
	`Router(config-if)# end`
	
	Example:
	`Router#`
		Returns the CLI to privileged EXEC mode.

Configuring and Enabling the Cross-Check Function

Perform this task to configure and enable cross-checking for an Up MEP. This task requires you to configure and enable cross-checking on two devices. This task is optional.

- Configuring and Enabling Cross-Checking for an Up MEP (U-PE A), page 118
- Configuring and Enabling Cross-Checking for an Up MEP (U-PE B), page 120
- Configuring and Enabling Cross-Checking for a Down MEP (CE-A), page 122
- Configuring and Enabling Cross-Checking for a Down MEP (CE-B), page 123

Configuring and Enabling Cross-Checking for an Up MEP (U-PE A)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*]
5. **exit**
6. **ethernet cfm mep crosscheck start-delay** *delay*
7. **exit**
8. **ethernet cfm mep crosscheck** {**enable** | **disable**} **domain** *domain-name* {**port** | **vlan**{*vlan-id* | *vlan-id* - *vlan-id* | **,** *vlan-id* - *vlan-id*}}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 4	**mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*] **Example:** `Router(config-ecfm)# mep crosscheck mpid 402 vlan 100`	Statically defines a remote MEP on a specified VLAN within the domain.

	Command or Action	Purpose

Step 5	**exit**
	
	Example:
	`Router(config-ecfm)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 6	**ethernet cfm mep crosscheck start-delay** *delay* **Example:** `Router(config)# ethernet cfm mep crosscheck start-delay 60`	Configures the maximum amount of time that the device waits for remote MEPs to come up before the cross-check operation is started

Step 7	**exit**
	
	Example:
	`Router(config)# exit`
	
	Example:
	`Router#`
		Returns the CLI to privileged EXEC mode.

Step 8	**ethernet cfm mep crosscheck** {**enable** | **disable**} **domain** *domain-name* {**port** | **vlan**{*vlan-id* | *vlan-id* **-** *vlan-id* | **,** *vlan-id* **-** *vlan-id*}} **Example:** `Router# ethernet cfm mep crosscheck enable domain cust4 vlan 100`	Enables cross-checking between the list of configured remote MEPs of a domain and MEPs learned through CCMs.

Examples

The following example configures cross-checking on an Up MEP (U-PE A):

```
U-PE A
ethernet cfm domain ServiceProvider level 4
mep mpid 402
!
ethernet cfm mep crosscheck start-delay 60
```

The following example enables cross-checking on an Up MEP (U-PE A):

```
U-PE A
U-PEA# ethernet cfm mep crosscheck enable domain cust4 vlan 100
```

Configuring and Enabling Cross-Checking for an Up MEP (U-PE B)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*]
5. **exit**
6. **ethernet cfm mep crosscheck start-delay** *delay*
7. **exit**
8. **ethernet cfm mep crosscheck** {**enable** | **disable**} **domain** *domain-name* {**port** | **vlan**{*vlan-id* | *vlan-id* - *vlan-id* | **,** *vlan-id* - *vlan-id*}}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 4	**mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*] **Example:** `Router(config-ecfm)# mep crosscheck mpid 401 vlan 100`	Statically defines a remote MEP on a specified VLAN within the domain.

	Command or Action	Purpose

Step 5	**exit**
	
	Example:
	`Router(config-ecfm)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 6	**ethernet cfm mep crosscheck start-delay** *delay* **Example:** `Router(config)# ethernet cfm mep crosscheck start-delay 60`	Configures the maximum amount of time that the device waits for remote MEPs to come up before the cross-check operation is started.

Step 7	**exit**
	
	Example:
	`Router(config)# exit`
	
	Example:
	`Router#`
		Returns the CLI to privileged EXEC mode.

Step 8	**ethernet cfm mep crosscheck** {**enable** | **disable**} **domain** *domain-name* {**port** | **vlan**{*vlan-id* | *vlan-id* **-** *vlan-id* | **,** *vlan-id* **-** *vlan-id*}} **Example:** `Router# ethernet cfm mep crosscheck enable domain cust4 vlan 100`	Enables cross-checking between the list of configured remote MEPs of a domain and MEPs learned through CCMs.

Examples

The following example configures cross-checking on an Up MEP (U-PE B):

```
U-PE B
ethernet cfm domain ServiceProvider level 4
mep mpid 401
!
ethernet cfm mep crosscheck start-delay 60
```

The following example enables cross-checking on an Up MEP (U-PE B):

```
U-PE B
U-PEB# ethernet cfm mep crosscheck enable domain cust4 vlan 100
```

Configuring and Enabling Cross-Checking for a Down MEP (CE-A)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep mpid** *mpid*
5. **exit**
6. **ethernet cfm mep crosscheck start-delay** *delay*
7. **exit**
8. **ethernet cfm mep crosscheck** {**enable** | **disable**} **domain** *domain-name* {**port** | **vlan**{*vlan-id* | *vlan-id* **-** *vlan-id* | **,** *vlan-id* **-** *vlan-id*}}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain Customer level 7`	Defines a CFM domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 4	**mep mpid** *mpid* **Example:** `Router(config-ecfm)# mep mpid 702`	Statically defines the MEPs within a maintenance association.

	Command or Action	Purpose

Step 5	**exit**
	
	Example:
	`Router(config-ecfm)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 6	**ethernet cfm mep crosscheck start-delay** *delay* **Example:** `Router(config)# ethernet cfm mep crosscheck start-delay 60`	Configures the maximum amount of time that the device waits for remote MEPs to come up before the cross-check operation is started.

Step 7	**exit**
	
	Example:
	`Router(config)# exit`
	
	Example:
	`Router#`
		Returns the CLI to privileged EXEC mode.

Step 8	**ethernet cfm mep crosscheck** {**enable** | **disable**} **domain** *domain-name* {**port** | **vlan**{*vlan-id* | *vlan-id* **-** *vlan-id* | **,** *vlan-id* **-** *vlan-id*}} **Example:** `Router# ethernet cfm mep crosscheck enable domain cust4 vlan 100`	Enables cross-checking between the list of configured remote MEPs of a domain and MEPs learned through CCMs.

Configuring and Enabling Cross-Checking for a Down MEP (CE-B)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep mpid** *mpid*
5. **exit**
6. **ethernet cfm mep crosscheck start-delay** *delay*
7. **exit**
8. **ethernet cfm mep crosscheck** {**enable** | **disable**} **domain** *domain-name* {**port** | **vlan**{*vlan-id* | *vlan-id* **-** *vlan-id* | **,** *vlan-id* **-** *vlan-id*}}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain Customer level 7`	Defines an outward CFM domain at a specified level and places the CLI in Ethernet CFM configuration mode.
Step 4	**mep mpid** *mpid* **Example:** `Router(config-ecfm)# mep mpid 702`	Statically defines the MEPs within a maintenance association.

	Command or Action	Purpose

Step 5	**exit**
	
	Example:
	`Router(config-ecfm)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 6	**ethernet cfm mep crosscheck start-delay** *delay* **Example:** `Router(config)# ethernet cfm mep crosscheck start-delay 60`	Configures the maximum amount of time that the device waits for remote MEPs to come up before the cross-check operation is started.

Step 7	**exit**
	
	Example:
	`Router(config)# exit`
	
	Example:
	`Router#`
		Returns the CLI to privileged EXEC mode.

Step 8	**ethernet cfm mep crosscheck** {**enable** | **disable**} **domain** *domain-name* {**port** | **vlan**{*vlan-id* | *vlan-id* **-** *vlan-id* | **,** *vlan-id* **-** *vlan-id*}} **Example:** `Router# ethernet cfm mep crosscheck enable domain cust4 vlan 100`	Enables cross-checking between the list of configured remote MEPs of a domain and MEPs learned through CCMs.

Configuring Ethernet OAM 802.3ah Interaction with CFM

For Ethernet OAM to function with CFM, you must configure an EVC and the OAM manager and associate the EVC with CFM. Additionally, you must use an Up MEP when you want interaction with the OAM manager.

- Configuring the OAM Manager, page 126
- Enabling Ethernet OAM, page 128

Configuring the OAM Manager

Note If you configure, change, or remove a UNI service type, EVC, Ethernet service instance, or CE-VLAN configuration, all configurations are checked to ensure that UNI service types are matched with EVC configurations and Ethernet service instances are matched with CE-VLAN configurations. Configurations are rejected if the pairings do not match.

Perform this task to configure the OAM manager on a PE device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]
5. **exit**
6. **exit**
7. **ethernet evc** *evc-id*
8. **oam protocol** {**cfm svlan** *svlan-id* **domain**
9. **exit**
10. Repeat Steps 3 through 9 to define other CFM domains that you want OAM manager to monitor.
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain cstmr1 level 3`	Defines a CFM domain, sets the domain level, and places the command-line interface (CLI) in Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 4	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**] **Example:** `Router(config-ecfm)# service vlan-id 10`	Configures a maintenance association within a maintenance domain and places the CLI into Ethernet CFM service configuration mode.

Step 5	**exit**
	
	Example:
	
	`Router(config-ecfm-srv)# exit`
	
	Example:
	
	`Router(config-ecfm)#`
		Returns the CLI to Ethernet CFM configuration mode.

Step 6	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 7	**ethernet evc** *evc-id* **Example:** `Router(config)# ethernet evc 50`	Defines an EVC and places the CLI in EVC configuration mode.
Step 8	**oam protocol** {**cfm svlan** *svlan-id* **domain** **Example:** *domain-name* | **ldp**} **Example:** `Router(config-evc)# oam protocol cfm svlan 10 domain cstmr1`	Configures the OAM protocol.

	Command or Action	Purpose

Step 9	**exit**
	
	Example:
	`Router(config-evc)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 10	Repeat Steps 3 through 9 to define other CFM domains that you want OAM manager to monitor.	--

Step 11	**end**
	
	Example:
	`Router(config)# end`
	
	Example:
	`Router#`
		Returns the CLI to privileged EXEC mode.

Enabling Ethernet OAM

The order in which the global and interface configuration commands are issued determines the configuration. The last command that is issued has precedence.

Perform this task to enable Ethernet OAM on a device or on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **switchport**
5. **ethernet oam** [**max-rate** *oampdus* | **min-rate** *num-seconds*| **mode** {**active** | **passive**} | **timeout** *seconds*]
6. **ethernet oam remote-loopback** {**supported** | **timeout** *seconds*}
7. **ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*}
8. **service instance** *id* **ethernet** [*evc-name*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1/3`	Specifies an interface and places the CLI in interface configuration mode.
Step 4	**switchport** **Example:** `Router(config-if)# switchport`	Configures a switchport.
Step 5	**ethernet oam** [**max-rate** *oampdus* | **min-rate** *num-seconds*| **mode** {**active** | **passive**} | **timeout** *seconds*] **Example:** `Router(config-if)# ethernet oam max-rate 50`	Enables Ethernet OAM on an interface.
Step 6	**ethernet oam remote-loopback** {**supported** | **timeout** *seconds*} **Example:** `Router(config-if)# ethernet oam remote-loopback supported`	Enables Ethernet remote loopback on the interface or sets a loopback timeout period.
Step 7	**ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep domain cstmr1 mpid 33 vlan 10`	Sets a port as internal to a maintenance domain and defines it as a MEP.

	Command or Action	Purpose
Step 8	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Router(config-if)# service instance 1 ethernet evc1`	Configures an Ethernet service instance and places the CLI in Ethernet CFM service configuration mode.

Step 9	**end**
	
	Example:
	
	`Router(config-ecfm-srv)# end`
	
	Example:
	
	`Router#`
		Returns the CLI to privileged EXEC mode.

Configuring CFM for Bridge Domains

Perform this task to configure Ethernet CFM for bridge domains. This task is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. Do one of the following:
   - **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]
5. **exit**
6. **exit**
7. **ethernet cfm domain** *domain-name* **level** *level-id*
8. **exit**
9. **ethernet cfm domain** *domain-name* **level** *level-id*
10. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]
11. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
12. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
13. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
14. **mep mpid** *mpid*
15. **exit**
16. **ethernet evc** *evc-name*
17. **exit**
18. **interface** *type number*
19. **no ip address**
20. **service instance** *id* **ethernet** [*evc-name*]
21. **encapsulation dot1q** *vlan-id*
22. **bridge-domain** *bridge-id*
23. **cfm mep domain** *domain-name* **mpid** *mpid-value*
24. **end**
25. **configure terminal**
26. **interface** *type name*
27. **no ip address**
28. **service instance** *id* **ethernet** [*evc-name*]
29. **encapsulation dot1q** *vlan-id*
30. **bridge-domain** *bridge-id*
31. **cfm mep domain** *domain-name* **mpid** *mpid-value*
32. **cfm mip level** *level-id*
33. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain CUSTOMER level 7`	Defines a CFM maintenance domain at a particular level and places the CLI in Ethernet CFM configuration mode.

Step 4	Do one of the following:
	
	• **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]
	
	Example:
	
	`Router(config-ecfm)# service s1 evc e1 vlan 10`
	
	Example:
	
	Example:
	
	`Router(config-ecfm)# service s1 evc e1`
		Configures a maintenance association within a maintenance domain and places the CLI into Ethernet CFM service configuration mode.

	Command or Action	Purpose

Step 5	**exit**
	
	Example:
	
	`Router(config-ecfm-srv)# exit`
	
	Example:
	
	`Router(config-ecfm)#`
		Returns the CLI to Ethernet CFM configuration mode.

Step 6	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 7	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain MIP level 7`	Defines a CFM maintenance domain at a particular level and places the CLI in Ethernet CFM configuration mode.

Step 8	**exit**
	
	Example:
	
	`Router(config-ecfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 9	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain PROVIDER level 4`	Defines a CFM maintenance domain at a particular level and places the CLI in Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 10	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**] **Example:** `Router(config-ecfm)# service vlan-id 10`	Configures a maintenance association within a maintenance domain and places the CLI into Ethernet CFM service configuration mode.
Step 11	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check interval 10s`	Enables the transmission of CCMs. • The time period between message transmissions is set.
Step 12	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check loss-threshold 5`	Enables the transmission of CCMs. • The number of CCMs missed before the remote MEP is declared down is set.
Step 13	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check static rmep`	Enables the transmission of CCMs. • Verification that the MEP received in the CCM is valid.
Step 14	**mep mpid** *mpid* **Example:** `Router(config-ecfm-srv)# mep mpid 200`	Statically defines MEPs within a maintenance association.

Step 15	**exit**
	
	Example:
	
	`Router(config-ecfm-srv)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 16	**ethernet evc** *evc-name* **Example:** `Router(config)# ethernet evc evc_100`	Defines an EVC and places the CLI in EVC configuration mode.

Step 17	**exit**
	
	Example:
	
	`Router(config-evc)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 18	**interface** *type number* **Example:** `Router(config)# interface Ethernet 1/0`	Specifies an interface and places the CLI in interface configuration mode.
Step 19	**no ip address** **Example:** `Router(config-if)# no ip address`	Disables IP processing.
Step 20	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Router(config-if)# service instance 100 ethernet evc_100`	Specifies an Ethernet service instance on an interface and places the CLI in service instance configuration mode.
Step 21	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to map 802.1Q frames on an ingress interface to the appropriate service instance.
Step 22	**bridge-domain** *bridge-id* **Example:** `Router(config-if-srv)# bridge-domain 100`	Establishes a bridge domain.

	Command or Action	Purpose
Step 23	**cfm mep domain** *domain-name* **mpid** *mpid-value* **Example:** `Router(config-if-srv)# cfm mep domain CUSTOMER mpid 1001`	Configures a MEP for a domain.

Step 24	**end**
	
	Example:
	
	`Router(config-if-srv)# end`
	
	Example:
	
	`Router#`
		Returns the CLI to privileged EXEC mode.

Step 25	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 26	**interface** *type name* **Example:** `Router(config)# interface Ethernet 1/1`	Specifies an interface and places the CLI in interface configuration mode.
Step 27	**no ip address** **Example:** `Router(config-if)# no ip address`	Disables IP processing.
Step 28	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Router(config-if)# service instance 100 ethernet evc_100`	Configures an Ethernet service instance on an interface and places the CLI in service instance configuration mode.
Step 29	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to map 802.1Q frames on an ingress interface to the appropriate service instance.

	Command or Action	Purpose
Step 30	**bridge-domain** *bridge-id* **Example:** `Router(config-if-srv)# bridge-domain 100`	Establishes a bridge domain.
Step 31	**cfm mep domain** *domain-name* **mpid** *mpid-value* **Example:** `Router(config-if-srv)# cfm mep domain PROVIDER mpid 201`	Configures a MEP for a domain.
Step 32	**cfm mip level** *level-id* **Example:** `Router(config-if-srv)# cfm mip level 4`	Configures a MIP at a specified level.

Step 33	**end**
	
	Example:
	
	`Router(config-if-srv)# end`
	
	Example:
	
	`Router#`
		Returns the CLI to privileged EXEC mode.

- Troubleshooting Tips, page 137

Troubleshooting Tips

To verify and isolate a fault, start at the highest level maintenance domain and do the following:

1. Check the device error status.
2. When a error exists, perform a loopback test to confirm the error.
3. Run a traceroute to the destination to isolate the fault.
4. If the fault is identified, correct the fault.
5. If the fault is not identified, go to the next lower maintenance domain and repeat steps 1 through 4 at that maintenance domain level.
6. Repeat the first four steps, as needed, to identify and correct the fault.

Configuration Examples for Configuring IEEE Ethernet CFM in a Service Provider Network

Example Provisioning a Network

This configuration example shows only CFM-related commands. All commands that are required to set up the data path and configure the VLANs on the device are not shown. However, it should be noted that CFM traffic will not flow into or out of the device if the VLANs are not properly configured.

```
CE-A
!
ethernet cfm global
ethernet cfm ieee
!
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
ethernet cfm mip auto-create level 7 vlan 1-4094
!
interface gigabitethernet3/2
 ethernet cfm mip level 7 vlan 101   <<<< Manual MIP
 ethernet cfm mep domain ServiceProvider-L4 mpid 401 vlan 101
 ethernet cfm mep domain OperatorA-L1 mpid 101 vlan 101
!
interface gigabitethernet4/2
 ethernet cfm mip level 1 vlan 101   <<<< Manual MIP
!
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
U-PE A
!
ethernet cfm global
ethernet cfm ieee
!
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
ethernet cfm global
ethernet cfm ieee
ethernet cfm domain OperatorA-L1 level 1
mep archive-hold-time 65
  mip auto-create
  service MetroCustomer1OpA vlan 101
!
interface gigabitethernet3/1
 ethernet cfm mip level 1 vlan 101   <<<< Manual MIP
!
interface gigabitethernet4/1
 ethernet cfm mip level 1    <<<< Manual MIP
!
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
PE-AGG A
ethernet cfm global
ethernet cfm ieee
ethernet cfm domain OperatorA-L1 level 1
mep archive-hold-time 65
  mip auto-create
  service MetroCustomer1OpA vlan 101
```

```
!
interface gigabitethernet3/1
 ethernet cfm mip level 1 vlan 101   <<<< Manual MIP
!
interface gigabitethernet4/1
 ethernet cfm mip level 1    <<<< Manual MIP
N-PE A
!
ethernet cfm global
ethernet cfm ieee
!
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
ethernet cfm domain ServiceProvider-L4 level 4
 mep archive-hold-time 60
 mip auto-create
 service MetroCustomer1 vlan 101
  continuity-check
!
ethernet cfm domain OperatorA level 1
mep archive-hold-time 65
 mip auto-create
service MetroCustomer1OpA vlan 101
  continuity-check
!
interface gigabitethernet3/0
 ethernet cfm mip level 1          <<<< manual MIP
!
interface gigabitethernet4/0
 ethernet cfm mip level 4    <<<< manual MIP
!
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
U-PE B
!
ethernet cfm global
ethernet cfm ieee
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
ethernet cfm domain Customer-L7 level 7
 mip auto-create
 service Customer1 vlan 101 direction down
!
ethernet cfm domain ServiceProvider-L4 level 4
 mep archive-hold-time 60
 service MetroCustomer1 vlan 101
  continuity-check
!
ethernet cfm domain OperatorB level 2
 mip auto-create
 mep archive-hold-time 65
 service MetroCustomer1OpB vlan 101
  continuity-check
!
interface gigabitethernet1/0
 ethernet cfm mip level 7   <<<< manual MIP
!
interface gigabitethernet2/0
 ethernet cfm mip level 2   <<<< manual MIP
!
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
PE-AGG B
ethernet cfm global
ethernet cfm ieee
!
ethernet cfm domain OperatorB level 2
 mep archive-hold-time 65
 mip auto-create
```

```
 service MetroCustomer1OpB vlan 101
!
interface gigabitethernet1/1
 ethernet cfm mip level 2   <<<< manual MIP
!
interface gigabitethernet2/1
 ethernet cfm mip level 2   <<<< manual MIP
N-PE B
!
ethernet cfm global
ethernet cfm ieee
!
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
ethernet cfm domain ServiceProvider level 4
 mep archive-hold-time 60
 mip auto-create
 service MetroCustomer1 vlan 101
  continuity-check
!
ethernet cfm domain OperatorB level 2
 mep archive-hold-time 65
 mip auto-create
 service MetroCustomer1OpB vlan 101
  continuity-check
!
interface gigabitethernet1/2
ethernet cfm mip level 2    <<<< manual MIP
!
interface gigabitethernet2/2
 ethernet cfm mip level 4   <<<< manual MIP
!
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
CE-B
!
ethernet cfm global
ethernet cfm ieee
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
ethernet cfm domain Customer-L7 level 7
 service Customer1 vlan 101 direction down
  continuity-check
!
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
```

Example Provisioning Service

```
CE-A
!
ethernet cfm global
ethernet cfm ieee
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
ethernet cfm domain Customer-L7 level 7
 service Customer1 vlan 101 direction down
  continuity-check
!
interface gigabitethernet3/2
 ethernet cfm mep domain Customer-L7 mpid 701 vlan 101
U-PE A
!
ethernet cfm global
ethernet cfm ieee
```

```
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
ethernet cfm mip auto-create level 7 vlan 1-4094
!
ethernet cfm domain ServiceProvider-L4 level 4
 mep archive-hold-time 60
 service MetroCustomer1 vlan 101
  continuity-check
!
ethernet cfm domain OperatorA-L1 level 1
 mep archive-hold-time 65
 mip auto-create
 service MetroCustomer1OpA vlan 101
  continuity-check
!
interface gigabitethernet3/2
 ethernet cfm mip level 7 vlan 101   <<<< Manual MIP
 ethernet cfm mep domain ServiceProvider-L4 mpid 401 vlan 101
 ethernet cfm mep domain OperatorA-L1 mpid 101 vlan 101
!
interface gigabitethernet4/2
 ethernet cfm mip level 1 vlan 101   <<<< Manual MIP
PE-AGG A
ethernet cfm global
ethernet cfm ieee
ethernet cfm domain OperatorA-L1 level 1
mep archive-hold-time 65
  mip auto-create
  service MetroCustomer1OpA vlan 101
!
interface gigabitethernet3/1
 ethernet cfm mip level 1 vlan 101   <<<< Manual MIP
!
interface gigabitethernet4/1
 ethernet cfm mip level 1    <<<< Manual MIP
N-PE A
!
ethernet cfm global
ethernet cfm ieee
!
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
ethernet cfm domain ServiceProvider-L4 level 4
 mep archive-hold-time 60
 mip auto-create
 service MetroCustomer1 vlan 101
  continuity-check
!
ethernet cfm domain OperatorA level 1
mep archive-hold-time 65
 mip auto-create
service MetroCustomer1OpA vlan 101
  continuity-check
!
interface gigabitethernet3/0
 ethernet cfm mip level 1    <<<< manual MIP
!
interface gigabitethernet4/0
 ethernet cfm mip level 4    <<<< manual MIP
 ethernet cfm mep domain OperatorA mpid 102 vlan 101
U-PE B
!
ethernet cfm global
ethernet cfm ieee
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
ethernet cfm domain Customer-L7 level 7
```

```
 mip auto-create
 service Customer1 vlan 101 direction down
!
ethernet cfm domain ServiceProvider-L4 level 4
 mep archive-hold-time 60
 service MetroCustomer1 vlan 101
  continuity-check
!
ethernet cfm domain OperatorB level 2
 mep archive-hold-time 65
 service MetroCustomer1OpB vlan 101
  continuity-check
!
interface gigabitethernet1/0
 ethernet cfm mip level 7   <<<< manual MIP
 ethernet cfm mep domain ServiceProvider-L4 mpid 402 vlan 101
 ethernet cfm mep domain OperatorB mpid 201 vlan 101
!
interface gigabitethernet2/0
 ethernet cfm mip level 2   <<<< manual MIP
N-PE B
!
ethernet cfm global
ethernet cfm ieee
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
ethernet cfm domain ServiceProvider level 4
 mep archive-hold-time 60
 mip auto-create
 service MetroCustomer1 vlan 101
  continuity-check
!
ethernet cfm domain OperatorB level 2
 mep archive-hold-time 65
 mip auto-create
 service MetroCustomer1OpB vlan 101
  continuity-check
!
interface gigabitethernet1/2
ethernet cfm mip level 2      <<<< manual MIP
!
interface gigabitethernet2/2
 ethernet cfm mip level 4     <<<< manual MIP
 ethernet cfm mep domain OperatorB mpid 202 vlan 101
CE-B
!
ethernet cfm global
ethernet cfm ieee
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
ethernet cfm domain Customer-L7 level 7
 service Customer1 vlan 101 direction down
  continuity-check
!
interface gigabitethernet3/2
 ethernet cfm mep domain Customer-L7 mpid 702 vlan 101
```

Additional References

Related Documents

Related Topic	Document Title
CFM commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Carrier Ethernet Command Reference*
Cisco IOS commands: master list of commands with complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS Master Command List, All Releases
Configuring Ethernet connectivity fault management in a service provider network (Cisco pre-Standard CFM Draft 1)	"Configuring Ethernet Connectivity Fault Management in a Service Provider Network" module in the *Cisco IOS Carrier Ethernet Configuration Guide*
Ethernet Local Management Interface on a provider edge device	"Configuring Ethernet Local Management Interface on a Provider Edge Device" module in the *Cisco IOS Carrier Ethernet Configuration Guide*
IP SLAs for Metro Ethernet	"IP SLAs for Metro Ethernet"
NSF/SSO and MPLS	"NSF/SSO - MPLS LDP and LDP Graceful Restart"
ISSU feature and functions	"Cisco IOS Broadband High Availability In Service Software Upgrade"
Performing an ISSU	"Cisco IOS In Service Software Upgrade Process and Enhanced Fast Software Upgrade Process"
SSO	"Stateful Switchover" chapter of the *Cisco IOS High Availability Configuration Guide*

Standards

Standard	Title
IEEE 802.1ag Standard	*802.1ag - Connectivity Fault Management*
IEEE 802.3ah	*IEEE 802.3ah Ethernet in the First Mile*
IETF VPLS OAM	*L2VPN OAM Requirements and Framework*
ITU-T	ITU-T Y.1731 OAM Mechanisms for Ethernet-Based Networks

MIBs

MIB	MIBs Link
CISCO-ETHER-CFM-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring IEEE Ethernet CFM in a Service Provider Network

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2 Feature Information for Configuring IEEE CFM in a Service Provider Network

Feature Name	Releases	Feature Information
802.1ag - IEEE D8.1 Standard-Compliant CFM, Y.1731 multicast LBM / AIS / RDI / LCK, IP SLA for Ethernet	12.2(33)SXI2 15.1(1)T	Ethernet CFM is an end-to-end per-service-instance Ethernet layer OAM protocol. CFM includes proactive connectivity monitoring, fault verification, and fault isolation for large Ethernet MANs and WANs. This feature is the implementation of IEEE 802.1ag Standard-Compliant CFM in Cisco IOS software. The following commands were introduced or modified: **alarm**, **clear ethernet cfm errors**, **clear ethernet cfm maintenance-points remote**, **clear ethernet cfm statistics**, **clear ethernet cfm traceroute-cache**, **continuity-check**, **cos**(CFM), **debug cfm**, **debug ethernet cfm all**, **debug ethernet cfm diagnostic**, **debug ethernet cfm error**, **debug ethernet cfm events**, **debug ethernet cfm ha**, **debug ethernet cfm packets**, **ethernet cfm alarm**, **ethernet cfm cc**, **ethernet cfm domain level**, **ethernet cfm global**, **ethernet cfm ieee**, **ethernet cfm interface**, **ethernet cfm logging**, **ethernet cfm mep crosscheck**, **ethernet cfm mep crosscheck start-delay**, **ethernet cfm mep domain mpid**, **ethernet cfm mip**, **ethernet cfm mip level**, **ethernet cfm traceroute cache**, **ethernet cfm traceroute cache hold-time**, **ethernet cfm traceroute cache size**, **id** (CFM), **maximum meps**, **mep archive-hold-time**, **mep mpid**, **mip auto-create**, **mip auto-create**(cfm-srv), **ping ethernet**, **sender-id**, **sender-id** (cfm-srv), **service**, **show ethernet cfm domain**, **show ethernet cfm errors**, **show ethernet cfm maintenance-**

Feature Name	Releases	Feature Information
		points local, **show ethernet cfm maintenance-points remote**, **show ethernet cfm maintenance-points remote detail**, **show ethernet cfm mpdb**, **show ethernet cfm statistics**, **show ethernet cfm traceroute-cache**, **snmp-server enable traps ethernet cfm cc**, **snmp-server enable traps ethernet cfm crosscheck**, **traceroute ethernet**.
IEEE 802.1ag-2007 Compliant CFM - Bridge Domain Support	12.2(33)SRE 12.2(50)SY	This feature provides support for bridge domains in IEEE 802.1ag Standard-Compliant CFM in Cisco IOS software. The following commands were introduced or modified: **cfm encapsulation**, **cfm mep domain**, **debug ethernet cfm all**, **debug ethernet cfm events**, **debug ethernet cfm packets**, **ethernet cfm mep crosscheck**, **service evc**, **show ethernet cfm maintenance-points remote crosscheck**, **show ethernet cfm maintenance-points remote detail**.

Glossary

CCM --continuity check message. A multicast CFM frame that a MEP transmits periodically to ensure continuity across the maintenance entities to which the transmitting MEP belongs, at the MA level on which the CCM is sent. No reply is sent in response to receiving a CCM.

configuration error list --Used to maintain a list of informational configuration errors for the port whenever a MEP is created or deleted. The information is displayed using the **show ethernet cfm** command

EVC --Ethernet virtual connection. An association of two or more user-network interfaces.

fault alarm --An out-of-band signal, typically an SNMP notification, that notifies a system administrator of a connectivity failure.

maintenance domain --The network or part of the network belonging to a single administration for which faults in connectivity are to be managed. The boundary of a maintenance domain is defined by a set of destination service access points (DSAPs), each of which may become a point of connectivity to a service instance.

maintenance domain name --The unique identifier of a domain that CFM is to protect against accidental concatenation of service instances.

MCL --maximum configured level. The highest level (0-7) service for Up MEPs, Down MEPs, or a MIP. This value is kept per service, either VLAN or bridge domain.

MEP --maintenance endpoint. An actively managed CFM entity associated with a specific DSAP of a service instance, which can generate and receive CFM frames and track any responses. It is an endpoint of a single MA, and terminates a separate maintenance entity for each of the other MEPs in the same MA.

MEP CCDB --A database, maintained by every MEP, that maintains received information about other MEPs in the maintenance domain.

MIP --maintenance intermediate point. A CFM entity, associated with a specific pair of ISS SAPs or EISS Service Access Points, which reacts and responds to CFM frames. It is associated with a single maintenance association and is an intermediate point within one or more maintenance entities.

MIP CCDB --A database of information about the MEPs in the maintenance domain. The MIP CCDB can be maintained by a MIP.

MP --maintenance point. Either a MEP or a MIP.

MPID --maintenance endpoint identifier. A small integer, unique over a given MA, that identifies a specific MEP.

OAM --operations, administration, and maintenance. A term used by several standards bodies to describe protocols and procedures for operating, administrating, and maintaining networks. Examples are ATM OAM and IEEE Std. 802.3ah OAM.

operator --Entity that provides a service provider a single network of provider bridges or a single Layer 2 or Layer 3 backbone network. An operator may be identical to or a part of the same organization as the service provider. For purposes of IEEE P802.1ag/D1.0, Draft Standard for Local and Metropolitan Area Networks, the operator and service provider are presumed to be separate organizations.

Terms such as “customer,” “service provider,” and “operator” reflect common business relationships among organizations and individuals that use equipment implemented in accordance with IEEE P802.1ag/D1.0.

UNI --user-network interface. A common term for the connection point between an operator’s bridge and customer equipment. A UNI often includes a C-VLAN-aware bridge component. The term UNI is used broadly in the IEEE P802.1ag/D1.0 standard when the purpose for various features of CFM are explained. UNI has no normative meaning.

Up MEP --A MEP that resides in a bridge and transmits to and receives CFM messages from the direction of the bridge relay entity.

Configuring Ethernet Connectivity Fault Management in a Service Provider Network

Ethernet Connectivity Fault Management (CFM) is an end-to-end per-service-instance Ethernet layer operations, administration, and maintenance (OAM) protocol. It includes proactive connectivity monitoring, fault verification, and fault isolation for large Ethernet metropolitan-area networks (MANs) and WANs.

The advent of Ethernet as a MAN and WAN technology imposes a new set of OAM requirements on Ethernet's traditional operations, which were centered on enterprise networks only. The expansion of Ethernet technology into the domain of service providers, where networks are substantially larger and more complex than enterprise networks and the user base is wider, makes operational management of link uptime crucial. More importantly, the timeliness in isolating and responding to a failure becomes mandatory for normal day-to-day operations, and OAM translates directly to the competitiveness of the service provider.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring Ethernet CFM in a Service Provider Network

Business Requirements

- Network topology and network administration have been evaluated.
- Business and service policies have been established.
- Partial Route Computation (PRC) codes have been implemented for all supported commands related to configuring High Availability (HA) on a maintenance endpoint (MEP), maintenance intermediate point (MIP), level, service instance ID, cross-check timer, cross-check, and domain.
- To use Non-Stop Forwarding (NSF) and In Service Software Upgrade (ISSU), Stateful Switchover (SSO) must be configured and working properly.

Restrictions for Configuring Ethernet CFM in a Service Provider Network

- In Cisco IOS releases earlier than Release 12.2(33)SRD, CFM and Per VLAN Spanning Tree (PVST) protocol cannot coexist on the same system.
- CFM cannot function when the following line cards are used on the same system:
    - FI_WS_X6196_RJ45
    - FI_WS_X6196_RJ21
    - FI_WS_X6548_RJ45
    - FI_WS_X6548_RJ21
- In Cisco IOS Release 12.2(33)SRD, support for the coexistence of CFM and PVST was introduced; however, for both protocols to function on the same system, each line card must support at least three match registers and at least one line card must be able to support only a 44-bit MAC match. The exception is the Cisco 7600 Series Supervisor Engine 720, which can support CFM/PVST coexistence with only two match registers.
- CFM loopback messages will not be confined within a maintenance domain according to their maintenance level. The impact of not having CFM loopback messages confined to their maintenance levels occurs at these levels:
    - Architecture--CFM layering is violated for loopback messages.
    - Deployment--A user may potentially misconfigure a network and have loopback messages succeed.
    - Security--A malicious device that recognizes devices’ MAC addresses and levels may potentially explore a network topology that should be transparent.
- Routed interfaces are supported only in Cisco IOS Release 12.4(11)T.
- CFM is not fully supported on a Multiprotocol Label Switching (MPLS) provider edge (PE) device. There is no interaction between CFM and an Ethernet over MPLS (EoMPLS) pseudowire. A CFM packet can be transparently passed like regular data packets only via pseudowire, with the following restrictions:
    - For Policy Feature Card (PFC)-based EoMPLS, which uses a Cisco Catalyst LAN card as the MPLS uplink port, a CFM packet can be transparently passed via an EoMPLS pseudowire like

regular data packets. The EoMPLS endpoint interface, however, cannot be a MEP or a MIP, although a CFM MEP or MIP can be supported on regular Layer 2 switchport interfaces.

- CFM configuration is not supported on an EtherChannel in FastEthernet Channel (FEC) mode.
- The Ethernet-OAM3.0: CFM Over BD, Untagged feature is supported only on ES20 and ES40 line cards.
- The HA features NFS/SSO Support in CFM 802.1ag/1.0d and ISSU Support in CFM 802.1ag/1.0d are not supported on customer edge (CE) devices.
- The NFS/SSO Support in CFM 802.1ag/1.0d feature is not supported for the traceroute and error databases.
- Cisco IOS Release 12.2(33)SRD does not support CFM messages passing through a blocked port.
- Cisco IOS Release 12.2(33)SXI1 does not support CFM.

Information About Configuring Ethernet CFM in a Service Provider Network

Ethernet CFM

Ethernet CFM is an end-to-end per-service-instance Ethernet layer OAM protocol that includes proactive connectivity monitoring, fault verification, and fault isolation. End to end can be PE to PE or CE to CE. A service can be identified as a service provider VLAN (S-VLAN) or an EVC service.

Being an end-to-end technology is the distinction between CFM and other metro-Ethernet OAM protocols. For example, MPLS, ATM, and SONET OAM help in debugging Ethernet wires but are not always end-to-end. 802.3ah OAM is a single-hop and per-physical-wire protocol. It is not end to end or service aware. Ethernet Local Management Interface (E-LMI) is confined between the user-end provider edge (uPE) and CE and relies on CFM for reporting status of the metro-Ethernet network to the CE.

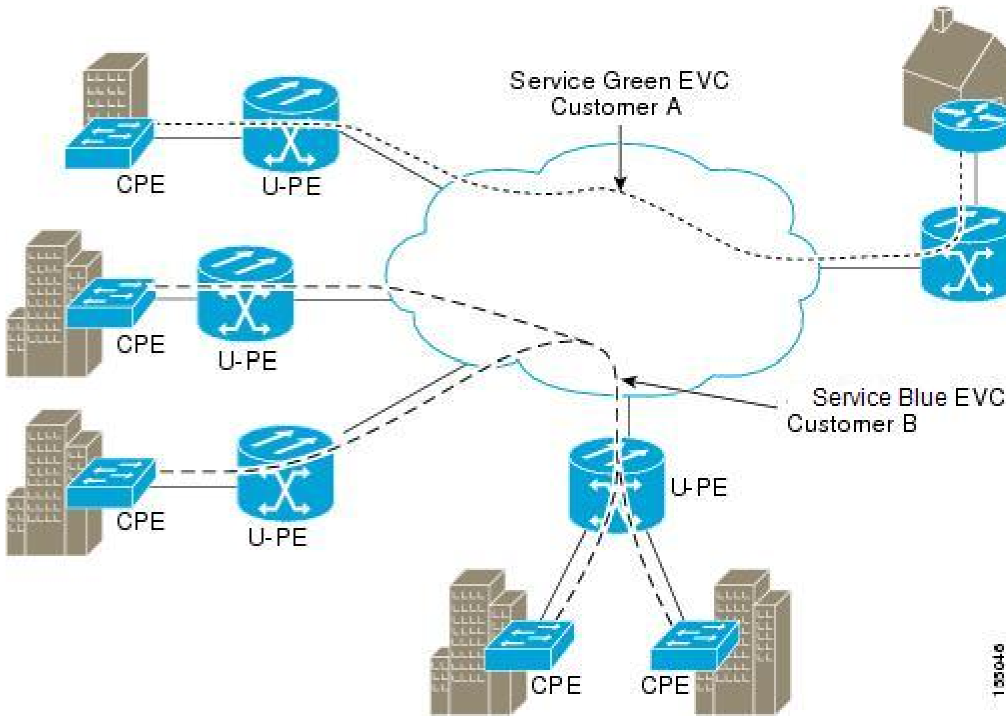
Troubleshooting carrier networks offering Ethernet Layer 2 services is challenging. Customers contract with service providers for end-to-end Ethernet service and service providers may subcontract with operators to provide equipment and networks. Compared to enterprise networks, where Ethernet traditionally has been implemented, these constituent networks belong to distinct organizations or departments, are substantially larger and more complex, and have a wider user base. Ethernet CFM provides a competitive advantage to service providers for which the operational management of link uptime and timeliness in isolating and responding to failures is crucial to daily operations.

Benefits of Ethernet CFM

- End-to-end service-level OAM technology
- Reduced operating expense for service provider Ethernet networks
- Competitive advantage for service providers
- Supports both distribution and access network environments with the outward facing MEPs enhancement

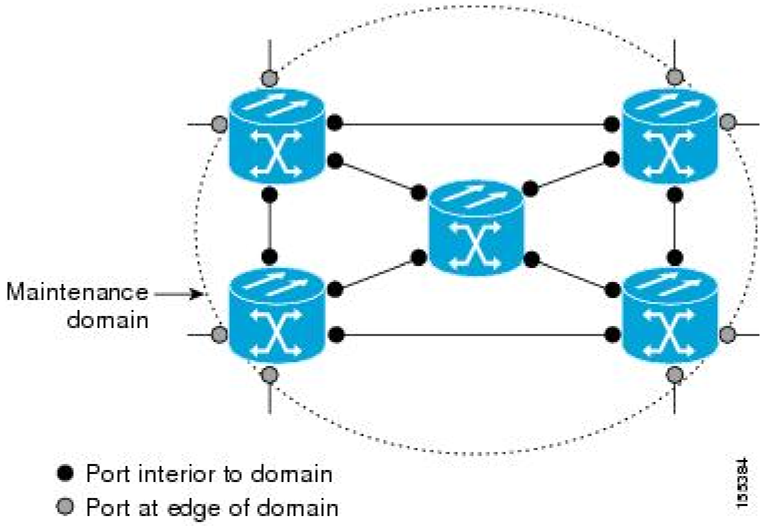
Customer Service Instance

A customer service instance is an Ethernet virtual connection (EVC), which is identified by an S-VLAN within an Ethernet island, and is identified by a globally unique service ID. A customer service instance can be point-to-point or multipoint-to-multipoint. The figure below shows two customer service instances. Service Instance Green is point to point; Service Instance Blue is multipoint to multipoint.



Maintenance Domain

A maintenance domain is 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 ports internal to it and at its boundary. The figure below illustrates a typical maintenance domain.

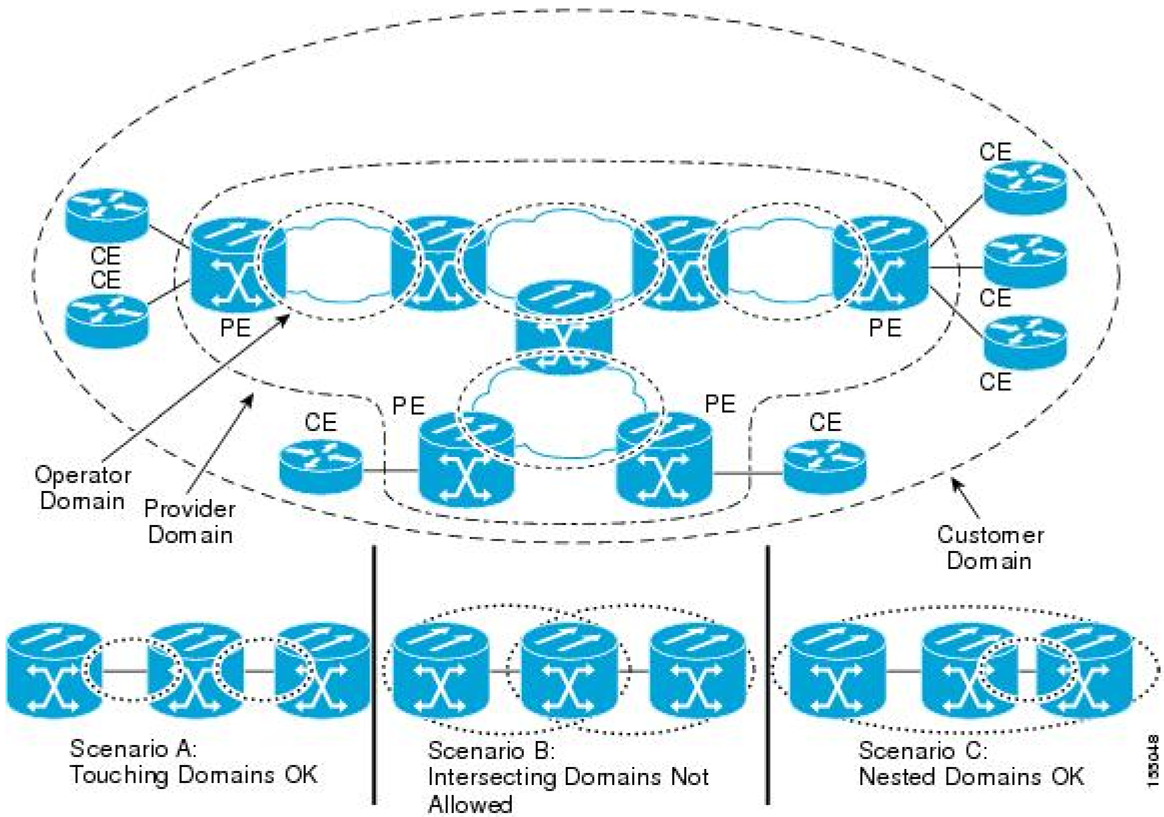


A unique maintenance level in the range of 0 to 7 is assigned to each domain by a network administrator. Levels and domain names are useful for defining the hierarchical relationship that exists among domains. The hierarchical relationship of domains parallels the structure of customer, service provider, and operator. The larger the domain, the higher the level value. For example, a customer domain would be larger than an operator domain. The customer domain may have a maintenance level of 7 and the operator domain may have a maintenance level of 0. Typically, operators would have the smallest domains and customers the largest domains, with service provider domains between them in size. All levels of the hierarchy must operate together.

Domains should not intersect because intersecting would mean management by more than one entity, which is not allowed. Domains may nest or touch but when two domains nest, the outer domain must have a higher maintenance level than the domain nested within it. Nesting maintenance domains is useful in the business model where a service provider contracts with one or more operators to provide Ethernet service to a customer. Each operator would have its own maintenance domain and the service provider would define its domain--a superset of the operator domains. Furthermore, the customer has its own end-to-end domain which is in turn a superset of the service provider domain. Maintenance levels of various nesting domains should be communicated among the administering organizations. For example, one approach would be to have the service provider assign maintenance levels to operators.

CFM exchanges messages and performs operations on a per-domain basis. For example, running CFM at the operator level does not allow discovery of the network by the higher provider and customer levels.

Network designers decide on domains and configurations. The figure below illustrates a hierarchy of operator, service provider, and customer domains and also illustrates touching, intersecting, and nested domains.



Maintenance Point

A maintenance point is a demarcation point on an interface (port) that participates in CFM within a maintenance domain. Maintenance points on device ports act as filters that confine CFM frames within the bounds of a domain by dropping frames that do not belong to the correct level. Maintenance points must be explicitly configured on Cisco devices. Two classes of maintenance points exist, MEPs and MIPs.

- Maintenance Endpoints, page 154
- Maintenance Intermediate Points, page 155

Maintenance Endpoints

MEPs have the following characteristics:

- Per maintenance domain (level) and service (S-VLAN or EVC)
- At the edge of a domain, define the boundary
- Within the bounds of a maintenance domain, confine CFM messages
- When configured to do so, proactively transmit CFM continuity check messages (CCMs)
- At the request of an administrator, transmit traceroute and loopback messages

Inward Facing MEPs

Inward facing means the MEP communicates through the Bridge Relay function and uses the Bridge-Brain MAC address. An inward facing MEP performs the following functions:

- Sends and receives CFM frames at its level through the relay function, not via the wire connected to the port on which the MEP is configured.
- Drops all CFM frames at its level (or lower level) that come from the direction of the wire.
- Processes all CFM frames at its level coming from the direction of the relay function.
- Drops all CFM frames at a lower level coming from the direction of the relay function.
- Transparently forwards all CFM frames at its level (or a higher level), independent of whether they come in from the relay function side or the wire side.

Note For the current Cisco IOS implementation, a MEP of level L (where L is less than 7) requires a MIP of level M > L on the same port; hence, CFM frames at a level higher than the level of the MEP will be catalogued by this MIP.

- If the port on which the inward MEP is configured is blocked by Spanning-Tree Protocol, the MEP can no longer transmit or receive CFM messages.

Outward Facing MEPs for Routed Ports and Switch Ports

Outward facing means that the MEP communicates through the wire. Outward facing MEPs can be configured on routed ports and switch ports. A MIP configuration at a level higher than the level of the outward facing MEP is not required.

Outward facing MEPs on routed ports use the port MAC address. Outward facing MEPs on port channels use the Bridge-Brain MAC address of the first member link. When port channel members change, the identities of outward facing MEPs do not have to change. Cisco IOS Release 12.2(33)SRD supports outward facing MEPs on switch ports and Ethernet flow points (EFPs).

An outward facing MEP performs the following functions:

- Sends and receives CFM frames at its level via the wire connected to the port where the MEP is configured.
- Drops all CFM frames at its level (or at a lower level) that come from the direction of the relay function.
- Processes all CFM frames at its level coming from the direction of the wire.
- Drops all CFM frames at a lower level coming from the direction of the wire.
- Transparently forwards all CFM frames at levels higher than the level of the outward facing MEP, independent of whether they come in from the relay function side or the wire side. This function is not applicable to routed ports.
- If the port on which the outward MEP is configured is blocked by Spanning-Tree Protocol, the MEP can still transmit and receive CFM messages via the wire. Cisco IOS Release 12.2(33)SRD does not support CFM messages passing through a blocked port.

Maintenance Intermediate Points

MIPs have the following characteristics:

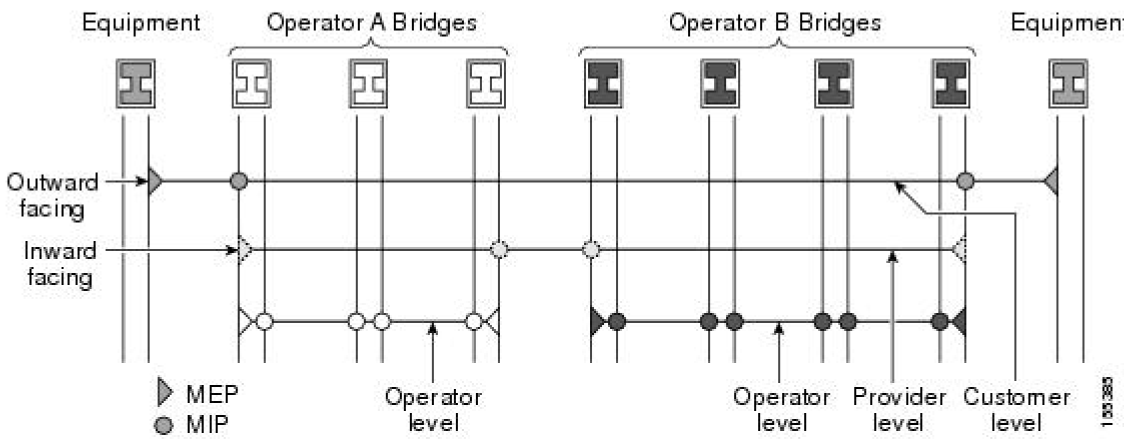
- Per maintenance domain (level) and for all S-VLANs enabled or allowed on a port.
- Internal to a domain, not at the boundary.
- CFM frames received from MEPs and other MIPs are cataloged and forwarded, using both the wire and the relay function.
- All CFM frames at a lower level are stopped and dropped, independent of whether they originate from the wire or relay function.

- All CFM frames at a higher level are forwarded, independent of whether they arrive from the wire or relay function.
- Passive points respond only when triggered by CFM traceroute and loopback messages.
- Bridge-Brain MAC addresses are used.

If the port on which a MIP is configured is blocked by Spanning-Tree Protocol, the MIP cannot receive CFM messages or relay them toward the relay function side. The MIP can, however, receive and respond to CFM messages from the wire.

A MIP has only one level associated with it and the command-line interface (CLI) does not allow you to configure a MIP for a domain that does not exist.

The figure below illustrates MEPs and MIPs at the operator, service provider, and customer levels.



CFM Messages

CFM uses standard Ethernet frames. CFM frames are distinguishable by EtherType and for multicast messages by MAC address. CFM frames are sourced, terminated, processed, and relayed by bridges. Routers can support only limited CFM functions.

Bridges that cannot interpret CFM messages forward them as normal data frames. All CFM messages are confined to a maintenance domain and to an S-VLAN (PE-VLAN or Provider-VLAN). Three types of messages are supported:

- Continuity Check
- Loopback
- Traceroute

Continuity Check Messages

CFM CCMs are multicast heartbeat messages exchanged periodically among MEPs. They allow MEPs to discover other MEPs within a domain and allow MIPs to discover MEPs. CCMs are confined to a domain and S-VLAN.

CFM CCMs have the following characteristics:

- Transmitted at a configurable periodic interval by MEPs. The interval can be from 10 seconds to 65535 seconds, the default is 30.
- Contain a configurable hold-time value to indicate to the receiver the validity of the message. The default is 2.5 times the transmit interval.

- Catalogued by MIPs at the same maintenance level.
- Terminated by remote MEPs at the same maintenance level.
- Unidirectional and do not solicit a response.
- Carry the status of the port on which the MEP is configured.

Loopback Messages

CFM loopback messages are unicast frames that a MEP transmits, at the request of an administrator, to verify connectivity to a particular maintenance point. A reply to a loopback message indicates whether a destination is reachable but does not allow hop-by-hop discovery of the path. A loopback message is similar in concept to an Internet Control Message Protocol (ICMP) Echo (ping) message.

A CFM loopback message can be generated on demand using the CLI. The source of a loopback message must be a MEP; the destination may be a MEP or a MIP. CFM loopback messages are unicast; replies to loopback messages also are unicast. CFM loopback messages specify the destination MAC address, VLAN, and maintenance domain.

Traceroute Messages

CFM traceroute messages are multicast frames that a MEP transmits, at the request of an administrator, to track the path (hop-by-hop) to a destination MEP. They allow the transmitting node to discover vital connectivity data about the path, and allow the discovery of all MIPs along the path that belong to the same maintenance domain. For each visible MIP, traceroute messages indicate ingress action, relay action, and egress action. Traceroute messages are similar in concept to User Datagram Protocol (UDP) traceroute messages.

Traceroute messages include the destination MAC address, VLAN, and maintenance domain and they have Time To Live (TTL) to limit propagation within the network. They can be generated on demand using the CLI. Traceroute messages are multicast; reply messages are unicast.

Cross-Check Function

The cross-check function is a timer-driven post-provisioning service verification between dynamically discovered MEPs (via CCMs) and expected MEPs (via configuration) for a service. The cross-check function verifies that all endpoints of a multipoint or point-to-point service are operational. The function supports notifications when the service is operational; otherwise it provides alarms and notifications for unexpected endpoints or missing endpoints.

The cross-check function is performed one time. You must initiate the cross-check function from the CLI every time you want a service verification.

SNMP Traps

The support provided by the Cisco IOS software implementation of CFM traps is Cisco proprietary information. MEPs generate two types of Simple Network Management Protocol (SNMP) traps, continuity check (CC) traps and cross-check traps.

CC Traps

- MEP up--Sent when a new MEP is discovered, the status of a remote port changes, or connectivity from a previously discovered MEP is restored after interruption.
- MEP down--Sent when a timeout or last gasp event occurs.
- Cross-connect--Sent when a service ID does not match the VLAN.
- Loop--Sent when a MEP receives its own CCMs.

- Configuration error--Sent when a MEP receives a continuity check with an overlapping MPID.

Cross-Check Traps

- Service up--Sent when all expected remote MEPs are up in time.
- MEP missing--Sent when an expected MEP is down.
- Unknown MEP--Sent when a CCM is received from an unexpected MEP.

Ethernet CFM and Ethernet OAM Interaction

To understand how CFM and OAM interact, you should understand the following concepts:

- Ethernet Virtual Circuit, page 158
- OAM Manager, page 158
- CFM over Bridge Domains, page 158

Ethernet Virtual Circuit

An EVC as defined by the Metro Ethernet Forum is a port-level point-to-point or multipoint-to-multipoint Layer 2 circuit. EVC status can be used by a CE device either to find an alternative path in to the service provider network or in some cases, to fall back to a backup path over Ethernet or over another alternative service such as Frame Relay or ATM.

OAM Manager

The OAM manager is an infrastructure element that streamlines interaction between OAM protocols. The OAM manager requires two interworking OAM protocols, in this case Ethernet CFM and Ethernet OAM. Interaction is unidirectional from the OAM manager to the CFM protocol and the only information exchanged is the user network interface (UNI) port status. Additional port status values available include

- REMOTE_EE--Remote excessive errors
- LOCAL_EE--Local excessive errors
- TEST--Either remote or local loopback

After CFM receives the port status, it communicates that status across the CFM domain.

CFM over Bridge Domains

The Ethernet OAM 3.0--CFM over BD, Untagged feature allows untagged CFM packets to be associated with a MEP. An incoming untagged customer CFM packet has an EtherType of CFM and is mapped to an EVC or bridge domain (BD) based on the encapsulation configured on the EFP. The EFP is configured specifically to recognize these untagged packets.

An EFP is a logical demarcation point of an EVC on an interface and can be associated with a bridge domain. The VLAN ID is used to match and map traffic to the EFP. VLAN IDs have local significance per port similar to ATM/FrameRelay virtual circuits. CFM is supported on a bridge domain associated with an EFP. The association between the bridge domain and the EFP allows CFM to use the encapsulation on the EFP. All EFPs in the same bridge domain form a broadcast domain. The bridge domain ID determines the broadcast domain.

The distinction between a VLAN port and the EFP is the encapsulation. VLAN ports use a default dot1q encapsulation. For EFPs, untagged, single tagged, and double tagged encapsulation exists with dot1q and

IEEE dot1ad EtherTypes. Different EFPs belonging to the same bridge domain can use different encapsulations.

Note The Ethernet OAM 3.0--CFM over BD, Untagged feature is supported only on ES20 and ES40 line cards.

HA Features Supported by CFM

In access and service provider networks using Ethernet technology, HA is a requirement, especially on Ethernet OAM components that manage EVC connectivity. End-to-end connectivity status information is critical and must be maintained on a hot standby route processor (RP).

Note A hot standby RP has the same software image as the active RP and supports synchronization of line card, protocol, and application state information between RPs for supported features and protocols.

End-to-end connectivity status is maintained on the CE, PE, and access aggregation PE (uPE) network nodes based on information received by protocols such as Ethernet LMI, CFM, and 802.3ah. This status information is used to either stop traffic or switch to backup paths when an EVC is down.

Every transaction involves either accessing or updating data among various databases. If the database is synchronized across active and standby modules, the modules are transparent to clients.

The Cisco IOS infrastructure provides various component application program interfaces (APIs) that help to maintain a hot standby RP. Metro Ethernet HA clients E-LMI, HA/ISSU, CFM HA/ISSU, and 802.3ah HA/ISSU interact with these components, update the database, and trigger necessary events to other components.

Benefits of CFM HA

- Elimination of network downtime for Cisco IOS software image upgrades, allowing for faster upgrades that result in higher availability than versions earlier than Cisco IOS Release 12.2(33)SRD.
- Elimination of resource scheduling challenges associated with planned outages and late night maintenance windows.
- Accelerated deployment of new services and applications and facilitation of faster implementation of new features, hardware, and fixes than versions earlier than Cisco IOS Release 12.2(33)SRD.
- Reduced operating costs due to outages while delivering higher service levels than versions earlier than Cisco IOS Release 12.2(33)SRD.
- CFM updates its databases and controls its own HA messaging and versioning, and this control facilitates maintenance.

- CFM HA in a Metro Ethernet Network, page 159

CFM HA in a Metro Ethernet Network

A standalone CFM implementation does not have explicit HA requirements. When CFM is implemented on a CE or PE with E-LMI, CFM must maintain the EVC state, which requires HA because the EVC state is critical in maintaining end-to-end connectivity. CFM configures the platform with maintenance level, domain, and maintenance point, learns the remote maintenance point information, and maps it to the appropriate EVC. CFM then aggregates data received from all remote ports and updates E-LMI; consequently HA requirements vary for CE and PE.

None of the protocols used in a Metro Ethernet Network (MEN) take action based on an EVC state, but a CE device that uses the E-LMI protocol and receives EVC information will stop sending traffic to the MEN when the EVC is down. When an EVC is down, the CE may also use a backup network, if available.

The CE receives the EVC ID, associated customer VLANs, UNI information, EVC state, and remote UNI ID and state from the MEN. The CE relies on the EVC state to send or stop traffic to the MEN via E-LMI.

The PE has EVC configuration and associated customer VLAN information and derives the EVC state and remote UNI from CFM. This information is sent to the CE using E-LMI.

Note PEs and CEs running 802.3ah OAM must maintain the port state so peers are not affected by a switchover. This information is also sent to remote nodes in CFM CC messages.

NSF SSO Support in CFM 802.1ag 1.0d

The redundancy configurations SSO and NSF are both supported in Ethernet CFM and are automatically enabled. A switchover from an active to a standby RP occurs when the active RP fails, is removed from the networking device, or is manually taken down for maintenance. NSF interoperates with the SSO feature to minimize network downtime following a switchover. The primary function of Cisco NSF is to continue forwarding IP packets following an RP switchover.

For detailed information about SSO, see the “Stateful Switchover” chapter of the *Cisco IOS High Availability Configuration Guide*. For detailed information about the NSF feature, see the “Cisco Nonstop Forwarding” chapter of the *Cisco IOS High Availability Configuration Guide*.

ISSU Support in CFM 802.1ag 1.0d

ISSU allows you to perform a Cisco IOS software upgrade or downgrade without disrupting packet flow. CFM performs a bulk update and a runtime update of the continuity check database to the standby RP, including adding, deleting, or updating a row. This checkpoint data requires ISSU capability to transform messages from one release to another. All the components that perform active RP to standby RP updates using messages require ISSU support.

ISSU is automatically enabled in CFM and lowers the impact that planned maintenance activities have on network availability by allowing software changes while the system is in service. For detailed information about ISSU, see the “Cisco IOS In Service Software Upgrade Process” chapter of the *Cisco IOS High Availability Configuration Guide*.

How to Set Up Ethernet CFM in a Service Provider Network

- Designing CFM Domains, page 44
- Configuring Ethernet CFM, page 163
- Configuring Ethernet OAM Interaction with CFM, page 235

Designing CFM Domains

Note To have an operator, service provider, or customer domain is optional. A network may have a single domain or multiple domains. The steps listed here show the sequence when all three types of domains will be assigned.

- Knowledge and understanding of the network topology.
- Understanding of organizational entities involved in managing the network; for example, operators, service providers, network operations centers (NOCs), and customer service centers.
- Understanding of the type and scale of services to be offered.
- Agreement by all organizational entities on the responsibilities, roles, and restrictions for each organizational entity.
- Determination of the number of maintenance domains in the network.
- Determination of the nesting and disjoint maintenance domains.
- Assignment of maintenance levels and names to domains based on agreement between the service provider and operator or operators.
- Determination of whether the domain should be inward or outward.

SUMMARY STEPS

1. Determine operator level MIPs.
2. Determine operator level MEPs.
3. Determine service provider MIPs.
4. Determine service provider MEPs.
5. Determine customer MIPs.
6. Determine customer MEPs.

DETAILED STEPS

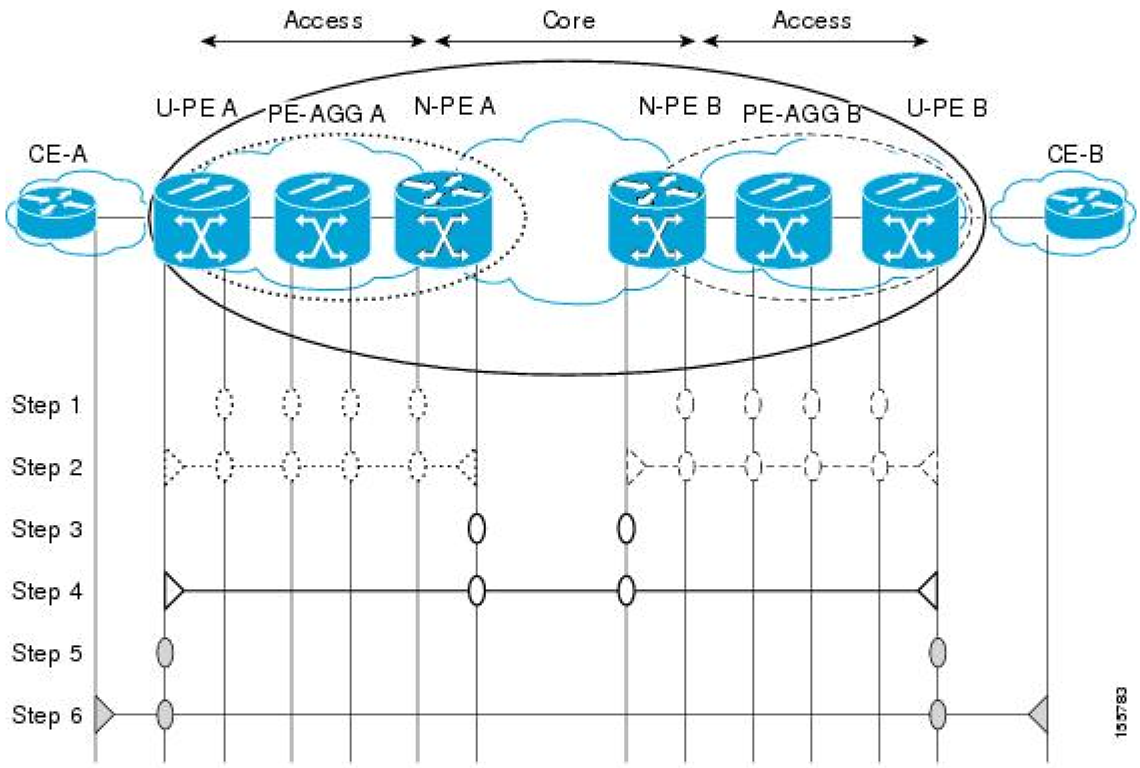
	Command or Action	Purpose
Step 1	Determine operator level MIPs.	Follow these steps: • Starting at lowest operator level domain, assign a MIP at every interface internal to the operator network to be visible to CFM. • Proceed to next higher operator level and assign MIPs. • Verify that every port that has a MIP at a lower level does not have maintenance points at a higher level. • Repeat steps a through d until all operator MIPs are determined.

	Command or Action	Purpose
Step 2	Determine operator level MEPs.	Follow these steps: • Starting at the lowest operator level domain, assign a MEP at every UNI that is part of a service instance. • Assign a MEP at the network to network interface (NNI) between operators, if there is more than one operator. • Proceed to next higher operator level and assign MEPs. • A port with a MIP at a lower level cannot have maintenance points at a higher level. A port with a MEP at a lower level should have either a MIP or MEP at a higher level.
Step 3	Determine service provider MIPs.	Follow these steps: • Starting at the lowest service provider level domain, assign service provider MIPs at the NNI between operators (if more than one). • Proceed to next higher service provider level and assign MIPs. • A port with a MIP at a lower level cannot have maintenance points at a higher level. A port with a MEP at a lower level should not have either a MIP or a MEP at a higher level.
Step 4	Determine service provider MEPs.	Follow these steps: • Starting at the lowest service provider level domain, assign a MEP at every UNI that is part of a service instance. • Proceed to next higher service provider level and assign MEPs. • A port with a MIP at a lower level cannot have maintenance points at a higher level. A port with a MEP at a lower level should have either a MIP or a MEP at a higher level.
Step 5	Determine customer MIPs.	Customer MIPs are allowed only on the UNIs at the uPEs if the service provider allows the customer to run CFM. Otherwise, the service provider can configure Cisco IOS devices to block CFM frames. • Configure a MIP on every uPE, at the UNI port, in the customer maintenance domain. • Ensure the MIPs are at a maintenance level that is at least one higher than the highest level service provider domain.
Step 6	Determine customer MEPs.	Customer MEPs are on customer equipment. Assign an outward facing MEP within an outward domain at the appropriate customer level at the handoff between the service provider and the customer.

- Examples, page 45
- Examples, page 45
- What to Do Next, page 163

Examples

The figure below shows an example of a network with a service provider and two operators, A and B. Three domains are to be established to map to each operator and the service provider. In this example, for simplicity we assume that the network uses Ethernet transport end to end. CFM, however, can be used with other transports.



What to Do Next

After you have defined the Ethernet CFM domains, configure Ethernet CFM functionality by first provisioning the network and then provisioning service.

Configuring Ethernet CFM

Configuring Ethernet CFM consists of the following tasks:

- Provisioning the Network, page 46
- Provisioning Service, page 76
- Configuring and Enabling the Cross-Check Function, page 221
- Configuring CFM over Bridge Domains, page 229
- Troubleshooting Tips, page 234

Provisioning the Network

- Provisioning the Network for CE-A, page 47
- Provisioning the Network for U-PE A, page 49
- Provisioning the Network for PE-AGG A, page 54
- Provisioning the Network for N-PE A, page 57
- Provisioning the Network for U-PE B, page 61
- Provisioning the Network for PE-AGG B, page 66

Provisioning the Network on the CE-A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id* **direction outward**
4. **mep archive-hold-time** *minutes*
5. **exit**
6. **ethernet cfm enable**
7. **ethernet cfm traceroute cache**
8. **ethernet cfm traceroute cache size** *entries*
9. **ethernet cfm traceroute cache hold-time** *minutes*
10. **ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* - *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* - *vlan-id*| [**,** *vlan-id* - *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*]
11. **snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**]
12. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**]
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **direction outward** **Example:** `Router(config)# ethernet cfm domain Customer level 7 direction outward`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 7	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 8	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 9	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.

	Command or Action	Purpose
Step 10	**ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets parameters for continuity check messages (CCMs).
Step 11	**snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM continuity check events.
Step 12	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up`	Enables SNMP trap generation for Ethernet CFM continuity check events in relation to the cross-check operation between statically configured MEPS and those learned via CCMs.
Step 13	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Provisioning the Network on the U-PE A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **ethernet cfm domain** *domain-name* **level** *level-id*
5. **mep archive-hold-time** *minutes*
6. **ethernet cfm domain** *domain-name* **level** *level-id*
7. **mep archive-hold-time** *minutes*
8. **exit**
9. **ethernet cfm enable**
10. **ethernet cfm traceroute cache**
11. **ethernet cfm traceroute cache size** *entries*
12. **ethernet cfm traceroute cache hold-time** *minutes*
13. **interface** *type number*
14. **ethernet cfm mip level** *level-id*
15. **exit**
16. **ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*]
17. **snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**]
18. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**]
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 4	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain at a particular maintenance level.
Step 5	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 6	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain OperatorA level 1`	Defines a domain.
Step 7	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 8	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 9	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 10	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.

	Command or Action	Purpose
Step 11	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 12	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 13	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet4/2`	Specifies an interface and enters interface configuration mode.
Step 14	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.
Step 16	**ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the parameters for CCMs.

	Command or Action	Purpose
Step 17	**snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.
Step 18	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up`	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.
Step 19	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Provisioning the Network on the PE-AGG A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **exit**
6. **ethernet cfm enable**
7. **interface** *type number*
8. **ethernet cfm mip level** *level-id*
9. **interface** *type number*
10. **ethernet cfm mip level** *level-id*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorA level 1`	Defines a domain and enters Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 7	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet3/1`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 8	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP on an interface.
Step 9	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet4/1`	Specifies an interface.
Step 10	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP on an interface.
Step 11	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Provisioning the Network on the N-PE A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **ethernet cfm domain** *domain-name* **level** *level-id*
6. **mep archive-hold-time** *minutes*
7. **exit**
8. **ethernet cfm enable**
9. **ethernet cfm traceroute cache**
10. **ethernet cfm traceroute cache size** *entries*
11. **ethernet cfm traceroute cache hold-time** *minutes*
12. **interface** *type number*
13. **ethernet cfm mip level** *level-id*
14. **exit**
15. **ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*]
16. **snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**]
17. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**]
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain and level and enters Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain OperatorA level 1`	Defines a domain and level.
Step 6	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 7	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 8	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 9	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 10	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.

	Command or Action	Purpose
Step 11	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 12	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet3/0`	Specifies an interface and enters interface configuration mode.
Step 13	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP on an interface.
Step 14	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.
Step 15	**ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the parameters for CCMs.
Step 16	**snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.

	Command or Action	Purpose
Step 17	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up`	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.
Step 18	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Provisioning the Network on the CE-B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
4. **mep archive-hold-time** *minutes*
5. **exit**
6. **ethernet cfm enable**
7. **ethernet cfm traceroute cache**
8. **ethernet cfm traceroute cache size** *entries*
9. **ethernet cfm traceroute cache hold-time** *minutes*
10. **ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*]
11. **snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**]
12. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**]
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain Customer level 7 direction outward`	Defines an outward CFM maintenance domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 7	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 8	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.

	Command or Action	Purpose
Step 9	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 10	**ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the parameters for CCMs.
Step 11	**snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.
Step 12	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up`	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.
Step 13	**end** **Example:** `Router(config)# end#`	Returns the CLI to privileged EXEC mode.

Provisioning the Network on the U-PE B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **ethernet cfm domain** *domain-name* **level** *level-id*
5. **mep archive-hold-time** *minutes*
6. **ethernet cfm domain** *domain-name* **level** *level-id*
7. **mep archive-hold-time** *minutes*
8. **exit**
9. **ethernet cfm enable**
10. **ethernet cfm traceroute cache**
11. **ethernet cfm traceroute cache size** *entries*
12. **ethernet cfm traceroute cache hold-time** *minutes*
13. **interface** *type number*
14. **ethernet cfm mip level** *level-id*
15. **exit**
16. **ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*]
17. **snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**]
18. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**]
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a specified level and enters Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 4	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain at a specified level.
Step 5	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 6	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain OperatorB level 2`	Defines a domain at a specified level.
Step 7	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 8	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 9	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 10	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.

	Command or Action	Purpose
Step 11	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 12	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 13	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet2/0`	Specifies an interface and enters interface configuration mode.
Step 14	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at a specified level on an interface.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.
Step 16	**ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the parameters for CCMs.

	Command or Action	Purpose
Step 17	**snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.
Step 18	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up`	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.
Step 19	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Provisioning the Network on the PE-AGG B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **exit**
6. **ethernet cfm enable**
7. **interface** *type number*
8. **ethernet cfm mip level** *level-id*
9. **interface** *type number*
10. **ethernet cfm mip level** *level-id*
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorB level 2`	Defines a domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 7	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet1/1`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 8	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at a specified level on an interface.
Step 9	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet2/1`	Specifies an interface.
Step 10	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at a specified level on an interface.
Step 11	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Provisioning the Network on the N-PE B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*]
4. **ethernet cfm domain** *domain-name* **level** *level-id*
5. **mep archive-hold-time** *minutes*
6. **ethernet cfm domain** *domain-name* **level** *level-id*
7. **mep archive-hold-time** *minutes*
8. **exit**
9. **ethernet cfm enable**
10. **ethernet cfm traceroute cache**
11. **ethernet cfm traceroute cache size** *entries*
12. **ethernet cfm traceroute cache hold-time** *minutes*
13. **interface** *type number*
14. **ethernet cfm mip level** *level-id*
15. **exit**
16. **snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**][**config**] [**loop**] [**cross-connect**]
17. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**]
18. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the parameters for CCMs and enters Ethernet CFM configuration mode.
Step 4	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain at a specified level and enters Ethernet CFM configuration mode.
Step 5	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 6	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain OperatorB level 2`	Defines a domain at a specified level.
Step 7	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 8	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 9	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 10	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 11	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 12	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 13	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet1/2`	Specifies an interface and enters interface configuration mode.
Step 14	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at a specified level on the interface.

Step 15	**exit**
	
	Example:
	
	`Router(config-if)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 16	**snmp-server enable traps ethernet cfm cc** [**mep-up**][**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.
Step 17	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown**| **mep-missing**| **service-up**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up`	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.
Step 18	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Provisioning Service

Provisioning Service on the CE-A

Perform this task to set up service for Ethernet CFM. Optionally, when this task is completed, you may configure and enable the cross-check function. To perform this optional task, see "Configuring and Enabling Cross-Checking for an Inward Facing MEP on the U PE-A".

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
4. **mep archive-hold-time** *minutes*
5. **service** *csi-id* **vlan** *vlan-id*
6. **exit**
7. **ethernet cfm enable**
8. **ethernet cfm traceroute cache**
9. **ethernet cfm traceroute cache size** *entries*
10. **ethernet cfm traceroute cache hold-time** *minutes*
11. **interface** *type number*
12. Do one of the following:
    - **ethernet cfm mep level** *level-id* [**inward**| **outward domain** *domain-name*] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
    -
    -
    - **switchport**
13. Do one of the following:
    - **interface** *type number* **.** *subinterface-number*
    -
    -
    - **switchport mode trunk**
14. Do one of the following:
    - **encapsulation dot1q** *vlan-id*
    -
    -
    - **ethernet cfm mep level** *level-id* [**inward**| **outward domain** *domain-name*] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain Customer level 7 direction outward`	Defines a CFM maintenance domain at a specified maintenance level and enters Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1 vlan 100`	Sets a universally unique ID for a CSI within the maintenance domain.
Step 6	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 7	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 8	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.

	Command or Action	Purpose
Step 9	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 10	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 11	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/3`	Specifies an interface and enters interface configuration mode.

Step 12	Do one of the following:
	• **ethernet cfm mep level** *level-id* [**inward**| **outward domain** *domain-name*] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
	•
	•
	• **switchport**
	
	Example:
	
	`Router(config-if)# ethernet cfm mep level 7 outward domain Customer mpid 701 vlan 100`
	
	Example:
	
	Example:
	
	Example:
	
	`Router(config-if)# switchport`
		Sets an interface as a domain boundary or specifies the interface type.

	Command or Action	Purpose

Step 13	Do one of the following:
	• **interface** *type number* **.** *subinterface-number*
	•
	•
	• **switchport mode trunk**
	
	Example:
	`Router(config-if)# interface ethernet 0/3.5`
	
	Example:
	
	Example:
	
	Example:
	`Router(config-if)# switchport mode trunk`
		Specifies a subinterface and enters subinterface configuration mode. The number that precedes the period (.) must match the number to which this subinterface belongs.
		
		Alternatively, specifies a trunking VLAN Layer 2 interface.

	Command or Action	Purpose

Step 14	Do one of the following:
	• **encapsulation dot1q** *vlan-id*
	•
	•
	• **ethernet cfm mep level** *level-id* [**inward**| **outward domain** *domain-name*] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
	Example:
	`Router(config-subif)# encapsulation dot1q 100`
	Example:
	Example:
	Example:
	`Router(config-if)# ethernet cfm mep level 7 outward domain Customer mpid 701 vlan 100`
		Enables IEEE 802.1Q encapsulation of traffic in a VLAN on a specified subinterface or provisions an interface as a domain boundary.

Step 15	**end** **Example:** `Router(config-if)# end#`	Returns the CLI to privileged EXEC mode.

Provisioning Service on the U-PE A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **ethernet cfm domain** *domain-name* **level** *level-id*
5. **mep archive-hold-time** *minutes*
6. **service** *csi-id* **vlan** *vlan-id*
7. **ethernet cfm domain** *domain-name* **level** *level-id*
8. **mep archive-hold-time** *minutes*
9. **service** *csi-id* **vlan** *vlan-id*
10. **exit**
11. **ethernet cfm enable**
12. **ethernet cfm traceroute cache**
13. **ethernet cfm traceroute cache size** *entries*
14. **ethernet cfm traceroute cache hold-time** *minutes*
15. **interface** *type number*
16. **ethernet cfm mip level** *level-id*
17. **ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
18. **ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
19. **interface** *type number*
20. **ethernet cfm mip level** *level-id*
21. **ethernet cfm cc enable level** {**any** | *level-id* | **,** *level-id*| *level-id* **-** *level-id*| **,** *level-id* **-** *level-id*} **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
22. **ethernet cfm cc enable level** {**any** | *level-id* | **,** *level-id*| *level-id* **-** *level-id*| **,** *level-id* **-** *level-id*} **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
23. **ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*]
24. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain at a specified level.
Step 5	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 6	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1 vlan 100`	Sets a universally unique ID on a VLAN for a CSI within the maintenance domain.
Step 7	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain OperatorA level 1`	Defines a domain at a specified level.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

	Command or Action	Purpose
Step 9	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1OpA vlan 100`	Sets a universally unique ID on a VLAN for a CSI within the maintenance domain.
Step 10	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 11	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 12	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 13	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 14	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 15	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet3/2`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 16	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 7`	Provisions a MIP at a specified maintenance level on the interface.
Step 17	**ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep level 4 mpid 401 vlan 100`	Provisions a MEP on the interface at a specified maintenance level and VLAN.
Step 18	**ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep level 1 mpid 101 vlan 100`	Provisions a MEP on the interface at a specified maintenance level and VLAN.
Step 19	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet 4/2`	Specifies an interface.
Step 20	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP on the interface at a specified maintenance level.
Step 21	**ethernet cfm cc enable level** {**any** | *level-id* | **,** *level-id*| *level-id* **-** *level-id*| **,** *level-id* **-** *level-id*} **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*} **Example:** `Router(config)# ethernet cfm cc enable level 4 vlan 100`	Globally enables transmission of CCMs at a specified level and VLAN.

	Command or Action	Purpose
Step 22	**ethernet cfm cc enable level** {**any** | *level-id* | **,** *level-id*| *level-id* **-** *level-id*| **,** *level-id* **-** *level-id*} **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*} **Example:** `Router(config)# ethernet cfm cc enable level 1 vlan 100`	Globally enables transmission of CCMs at a specified level and VLAN.
Step 23	**ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the parameters for CCMs.
Step 24	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Provisioning Service on the PE-AGG A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **service** *csi-id* **vlan** *vlan-id*
6. **exit**
7. **ethernet cfm enable**
8. **interface** *type number*
9. **ethernet cfm mip level** *level-id*
10. **interface** *type number*
11. **ethernet cfm mip level** *level-id*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorA level 1`	Defines a domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1OpA vlan 100`	Sets a universally unique ID on a specified VLAN for a CSI within the maintenance domain.
Step 6	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 7	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 8	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet3/1`	Specifies an interface and enters interface configuration mode.
Step 9	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP at a specified maintenance level on the interface.
Step 10	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet4/1`	Specifies an interface.
Step 11	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP at a specified maintenance level on the interface.
Step 12	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Provisioning Service on the N-PE A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **service** *csi-id* **vlan** *vlan-id*
6. **ethernet cfm domain** *domain-name* **level** *level-id*
7. **mep archive-hold-time** *minutes*
8. **service** *csi-id* **vlan** *vlan-id*
9. **exit**
10. **ethernet cfm enable**
11. **ethernet cfm traceroute cache**
12. **ethernet cfm traceroute cache size** *entries*
13. **ethernet cfm traceroute cache hold-time** *minutes*
14. **interface** *type number*
15. **ethernet cfm mip level** *level-id*
16. **interface** *type number*
17. **ethernet cfm mip level** *level-id*
18. **ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
19. **exit**
20. **ethernet cfm cc enable level** {**any** | *level-id* | **,** *level-id*| *level-id* **-** *level-id*| **,** *level-id* **-** *level-id*} **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
21. **ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*]
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1 vlan 100`	Sets a universally unique ID on a specified VLAN for a CSI within the maintenance domain.
Step 6	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain OperatorA level 1`	Defines a domain at a specified level.
Step 7	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 8	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1OpA vlan 100`	Sets a universally unique ID on a specified VLAN for a CSI within the maintenance domain.
Step 9	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 10	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 11	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 13	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet3/0`	Specifies an interface and enters interface configuration mode.
Step 15	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP at a specified maintenance level on the interface.
Step 16	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet4/0`	Specifies an interface.

	Command or Action	Purpose
Step 17	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 4`	Provisions a MIP at a specified maintenance level on the interface.
Step 18	**ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep level 2 mpid 102 vlan 100`	Sets the interface as a domain boundary (edge) at a specified level, defines a MEP, and specifies the VLAN.

Step 19	**exit**
	
	Example:
	
	`Router(config-if)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 20	**ethernet cfm cc enable level** {**any** | *level-id* | **,** *level-id*| *level-id* **-** *level-id*| **,** *level-id* **-** *level-id*} **vlan** {**any** | *vlan-id* | **,** *vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*} **Example:** `Router(config)# ethernet cfm cc enable level 1 vlan 100`	Globally enables transmission of CCMs at a specified level and VLAN.
Step 21	**ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the parameters for CCMs.
Step 22	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Provisioning Service on the CE-B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
4. **mep archive-hold-time** *minutes*
5. **service** *csi-id* **vlan** *vlan-id*
6. **exit**
7. **ethernet cfm enable**
8. **ethernet cfm traceroute cache**
9. **ethernet cfm traceroute cache size** *entries*
10. **ethernet cfm traceroute cache hold-time** *minutes*
11. **interface** *type number*
12. Do one of the following:
    - **ethernet cfm mep level** *level-id* [**inward**| **outward domain** *domain-name*] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
    -
    -
    - **switchport**
13. Do one of the following:
    - **interface** *type number* **.** *subinterface-number*
    -
    -
    - **switchport mode trunk**
14. Do one of the following:
    - **encapsulation dot1q** *vlan-id*
    -
    -
    - **ethernet cfm mep level** *level-id* [**inward**| **outward domain** *domain-name*] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
15. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain Customer level 7 direction outward`	Defines a CFM maintenance domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1 vlan 100`	Sets a universally unique ID for a CSI within a maintenance domain.

Step 6	**exit**
	
	Example:
	
	`Router(config-ether-cfm)# exit`
	
	Example:
	
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 7	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 8	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 9	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 10	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 11	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/1`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose

Step 12	Do one of the following:
	• **ethernet cfm mep level** *level-id* [**inward**| **outward domain** *domain-name*] **mpid** *id* **vlan** {**any** | *vlan-id* | ,*vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
	•
	•
	• **switchport**
	
	Example:
	`Router(config-if)# ethernet cfm mep level 7 outward domain Customer mpid 701 vlan 100`
	
	Example:
	
	Example:
	
	Example:
	`Router(config-if)# switchport`
		Sets an interface as a domain boundary or specifies the interface type.

	Command or Action	Purpose

Step 13	Do one of the following:
	• **interface** *type number* **.** *subinterface-number*
	•
	•
	• **switchport mode trunk**
	
	Example:
	`Router(config-if)# interface ethernet 0/3.5`
	
	Example:
	
	Example:
	
	Example:
	`Router(config-if)# switchport mode trunk`
		Specifies a subinterface and enters subinterface configuration mode. The number that precedes the period (.) must match the number to which this subinterface belongs.
		Alternatively, specifies a trunking VLAN Layer 2 interface.

	Command or Action	Purpose

Step 14	Do one of the following:
	• **encapsulation dot1q** *vlan-id*
	•
	•
	• **ethernet cfm mep level** *level-id* [**inward**| **outward domain** *domain-name*] **mpid** *id* **vlan** {**any** | *vlan-id* | ,*vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
	
	Example:
	`Router(config-subif)# encapsulation dot1q 100`
	
	Example:
	
	Example:
	
	Example:
	`Router(config-if)# ethernet cfm mep level 7 outward domain Customer mpid 701 vlan 100`
		Enables IEEE 802.1Q encapsulation of traffic in a VLAN on a specified subinterface or provisions an interface as a domain boundary.

Step 15	**end** **Example:** `Router(config-subif)# end`	Returns the CLI to privileged EXEC mode.

Provisioning Service on the U-PE B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **ethernet cfm domain** *domain-name* **level** *level-id*
5. **mep archive-hold-time** *minutes*
6. **service** *csi-id* **vlan** *vlan-id*
7. **ethernet cfm domain** *domain-name* **level** *level-id*
8. **mep archive-hold-time** *minutes*
9. **service** *csi-id* **vlan** *vlan-id*
10. **exit**
11. **ethernet cfm enable**
12. **ethernet cfm traceroute cache**
13. **ethernet cfm traceroute cache size** *entries*
14. **ethernet cfm traceroute cache hold-time** *minutes*
15. **interface** *type number*
16. **ethernet cfm mip level** *level-id*
17. **ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| *,vlan-id* **-** *vlan-id*}
18. **ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| *,vlan-id* **-** *vlan-id*}
19. **interface** *type number*
20. **ethernet cfm mip level** *level-id*
21. **exit**
22. **ethernet cfm cc enable level** {**any** | *level-id* | *,level-id*| *level-id* **-** *level-id*| **,** *level-id* **-** *level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| *,vlan-id* **-** *vlan-id*}
23. **ethernet cfm cc enable level** {**any** | *level-id* | *,level-id*| *level-id* **-** *level-id*| **,** *level-id* **-** *level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| *,vlan-id* **-** *vlan-id*}
24. **ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*]
25. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain at a specified level.
Step 5	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 6	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1 vlan 100`	Sets a universally unique ID on a specified VLAN for a CSI within the maintenance domain.
Step 7	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain OperatorB level 2`	Defines a domain at a specified level.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

	Command or Action	Purpose
Step 9	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1OpB vlan 100`	Sets a universally unique ID on a specified VLAN for a CSI within the maintenance domain.
Step 10	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 11	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 12	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 13	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 14	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 15	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet1/0`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 16	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 7`	Provisions a MIP at a specified maintenance level on the interface.
Step 17	**ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| *,vlan-id* **-** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep level 2 mpid 402 vlan 100`	Sets the interface as a domain boundary (edge) at a specified level, defines it as a MEP, and specifies the VLAN.
Step 18	**ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| *,vlan-id* **-** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep level 2 mpid 201 vlan 100`	Sets the interface as a domain boundary (edge) at a specified level, defines it as a MEP, and specifies the VLAN.
Step 19	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet2/0`	Specifies an interface.
Step 20	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at a specified maintenance level on the interface.
Step 21	**exit** **Example:** `Router(config-if)# exit#`	Returns the CLI to global configuration mode.
Step 22	**ethernet cfm cc enable level** {**any** | *level-id* | *,level-id*| *level-id* **-** *level-id*| **,** *level-id* **-** *level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| *,vlan-id* **-** *vlan-id*} **Example:** `Router(config)# ethernet cfm cc enable level 4 vlan 100`	Globally enables transmission of CCMs at a specified level and VLAN.

	Command or Action	Purpose
Step 23	**ethernet cfm cc enable level** {**any** | *level-id* | *,level-id*| *level-id* **-** *level-id*| **,** *level-id* **-** *level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| *,vlan-id* **-** *vlan-id*} **Example:** `Router(config)# ethernet cfm cc enable level 2 vlan 100`	Globally enables transmission of CCMs at a specified level and VLAN.
Step 24	**ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the parameters for CCMs.
Step 25	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Provisioning Service on the PE-AGG B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **service** *csi-id* **vlan** *vlan-id*
6. **exit**
7. **ethernet cfm enable**
8. **interface** *type number*
9. **ethernet cfm mip level** *level-id*
10. **interface** *type number*
11. **ethernet cfm mip level** *level-id*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain OperatorB level 2`	Defines a domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Set the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1OpB vlan 100`	Sets a universally unique ID on a specified VLAN for a CSI within the maintenance domain.
Step 6	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 7	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 8	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet1/1`	Specifies an interface and enters interface configuration mode.
Step 9	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at a specific maintenance level on an interface.
Step 10	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet2/1`	Specifies an interface.
Step 11	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at a specified maintenance level on the interface.
Step 12	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Provisioning Service on the N-PE B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep archive-hold-time** *minutes*
5. **service** *csi-id* **vlan** *vlan-id*
6. **ethernet cfm domain** *domain-name* **level** *level-id*
7. **mep archive-hold-time** *minutes*
8. **service** *csi-id* **vlan** *vlan-id*
9. **exit**
10. **ethernet cfm enable**
11. **ethernet cfm traceroute cache**
12. **ethernet cfm traceroute cache size** *entries*
13. **ethernet cfm traceroute cache hold-time** *minutes*
14. **interface** *type number*
15. **ethernet cfm mip level** *level-id*
16. **interface** *type number*
17. **ethernet cfm mip level** *level-id*
18. **ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
19. **exit**
20. **ethernet cfm cc enable level** {**any** | *level-id* | *,level-id*| *level-id* **-** *level-id*| **,** *level-id* **-** *level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}
21. **ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*]
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 5	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1 vlan 100`	Sets a universally unique ID on a specified VLAN for a CSI within the maintenance domain.
Step 6	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain OperatorB level 2`	Defines a domain at a specified level.
Step 7	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 8	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1OpB vlan 100`	Sets a universally unique ID for a CSI on a specified VLANwithin the maintenance domain.
Step 9	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 10	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 11	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.
Step 13	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet1/2`	Specifies an interface and enters interface configuration mode.
Step 15	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at a specified maintenance level on the interface.
Step 16	**interface** *type number* **Example:** `Router(config-if)# interface gigabitethernet2/2`	Specifies an interface.

	Command or Action	Purpose
Step 17	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 4`	Provisions a MIP at a specific maintenance level on an interface.
Step 18	**ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep level 2 mpid 202 vlan 100`	Sets the interface as a domain boundary (edge), defines it as a MEP, and specifies a VLAN.
Step 19	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.
Step 20	**ethernet cfm cc enable level** {**any** | *level-id* | *,level-id*| *level-id* **-** *level-id*| **,** *level-id* **-** *level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*} **Example:** `Router(config)# ethernet cfm cc enable level 2 vlan 100`	Globally enables transmission of CCMs at a specified level and VLAN.
Step 21	**ethernet cfm cc level** {**any** | *level-id* | *level-id* **-** *level-id*|[**,** *level-id* **-** *level-id*]} **vlan** {*vlan-id* | **any**| *vlan-id* **-** *vlan-id*| [**,** *vlan-id* **-** *vlan-id*]} [**interval** *seconds*] [**loss-threshold** *num-msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the parameters for CCMs.
Step 22	**end** **Example:** `Router(config)#`	Returns the CLI to privileged EXEC mode.

Configuring and Enabling the Cross-Check Function

- Configuring and Enabling Cross-Checking for an Inward Facing MEP on the U PE-A, page 222

- Configuring and Enabling Cross-Checking for an Inward Facing MEP on the U PE-B, page 224
- Configuring and Enabling Cross-Checking for an Outward Facing MEP on the CE-A, page 225
- Configuring and Enabling Cross-Checking for an Outward Facing MEP on the CE-B, page 227

Configuring and Enabling Cross-Checking for an Inward Facing MEP on the U PE-A

Perform this task to configure and enable cross-checking for an inward facing MEP. This task requires you to configure and enable cross-checking on two devices. This task is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*]
5. **exit**
6. **ethernet cfm mep crosscheck start-delay** *delay*
7. **exit**
8. **ethernet cfm mep crosscheck** {**enable** | **disable**} **level** {*level-id* | *level-id-level-id* [,*level-id-level-id*]} **vlan** {*vlan-id* | **any** | *vlan-id-vlan-id* [,*vlan-id-vlan-id*]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM domain at a specified level and enters Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 4	**mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*] **Example:** `Router(config-ether-cfm)# mep crosscheck mpid 402 vlan 100`	Statically defines a remote MEP on a specified VLAN within the domain.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit#`	Returns the CLI to global configuration mode.
Step 6	**ethernet cfm mep crosscheck start-delay** *delay* **Example:** `Router(config)# ethernet cfm mep crosscheck start-delay 60`	Configures the maximum amount of time that the device waits for remote MEPs to come up before the cross-check operation is started
Step 7	**exit** **Example:** `Router(config)# exit`	Returns the CLI to privileged EXEC mode.
Step 8	**ethernet cfm mep crosscheck** {**enable** | **disable**} **level** {*level-id* | *level-id-level-id* [,*level-id-level-id*]} **vlan** {*vlan-id* | **any** | *vlan-id-vlan-id* [,*vlan-id-vlan-id*]} **Example:** `Router# ethernet cfm mep crosscheck enable level 4 vlan 100`	Enables cross-checking between remote MEPs in the domain and MEPs learned through CCMs.

Example

The following example configures cross-checking on an inward facing MEP (U-PE A):

```
U-PE A
ethernet cfm domain ServiceProvider level 4
mep crosscheck mpid 402 vlan 100
!
ethernet cfm mep crosscheck start-delay 60
```

The following example enables cross-checking on an inward facing MEP (U-PE A):

```
U-PE A
U-PEA# ethernet cfm mep crosscheck enable level 4 vlan 100
```

Configuring and Enabling Cross-Checking for an Inward Facing MEP on the U PE-B

Perform this task to configure and enable cross-checking for an inward facing MEP. This task requires you to configure and enable cross-checking on two devices. This task is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*]
5. **exit**
6. **ethernet cfm mep crosscheck start-delay** *delay*
7. **exit**
8. **ethernet cfm mep crosscheck** {**enable** | **disable**} **level** {*level-id* | *level-id-level-id* [,*level-id-level-id*]} **vlan** {*vlan-id* | **any** | *vlan-id-vlan-id* [,*vlan-id-vlan-id*]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	**mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*] **Example:** `Router(config-ether-cfm)# mep crosscheck mpid 401 vlan 100`	Statically defines a remote MEP on a specified VLAN within the domain.

	Command or Action	Purpose
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet cfm mep crosscheck start-delay** *delay* **Example:** `Router(config)# ethernet cfm mep crosscheck start-delay 60`	Configures the maximum amount of time that the device waits for remote MEPs to come up before the cross-check operation is started.
Step 7	**exit** **Example:** `Router(config)# exit`	Returns the CLI to privileged EXEC mode.
Step 8	**ethernet cfm mep crosscheck** {**enable** | **disable**} **level** {*level-id* | *level-id-level-id* [,*level-id-level-id*]} **vlan** {*vlan-id* | **any** | *vlan-id-vlan-id* [,*vlan-id-vlan-id*]} **Example:** `Router# ethernet cfm mep crosscheck enable level 4 vlan 100`	Enables cross-checking between MEPs.

Example

The following example configures cross-checking on an inward facing MEP (U-PE B)

```
U-PE B
ethernet cfm domain ServiceProvider level 4
mep crosscheck mpid 401 vlan 100
!
ethernet cfm mep crosscheck start-delay 60
```

The following example enables cross-checking on an inward facing MEP (U-PE B)

```
U-PE B
U-PEB# ethernet cfm mep crosscheck enable level 4 vlan 100
```

Configuring and Enabling Cross-Checking for an Outward Facing MEP on the CE-A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
4. **mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*]
5. **exit**
6. **ethernet cfm mep crosscheck start-delay** *delay*
7. **exit**
8. **ethernet cfm mep crosscheck** {**enable** | **disable**} **level** {*level-id* | *level-id-level-id* [,*level-id-level-id*]} **vlan** {*vlan-id* | **any** | *vlan-id-vlan-id* [,*vlan-id-vlan-id*]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain Customer level 7 direction outward`	Defines a CFM domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	**mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*] **Example:** `Router(config-ether-cfm)# mep crosscheck mpid 702 vlan 100`	Statically defines a remote MEP with a specified ID, VLAN, and domain.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 6	**ethernet cfm mep crosscheck start-delay** *delay* **Example:** `Router(config)# ethernet cfm mep crosscheck start-delay 60`	Configures the maximum amount of time that the device waits for remote MEPs to come up before the cross-check operation is started.
Step 7	**exit** **Example:** `Router(config)# exit`	Returns the CLI to privileged EXEC mode.
Step 8	**ethernet cfm mep crosscheck** {**enable** | **disable**} **level** {*level-id* | *level-id-level-id* [,*level-id-level-id*]} **vlan** {*vlan-id* | **any** | *vlan-id-vlan-id* [,*vlan-id-vlan-id*]} **Example:** `Router# ethernet cfm mep crosscheck enable level 7 vlan 100`	Enables cross-checking between MEPs.

Configuring and Enabling Cross-Checking for an Outward Facing MEP on the CE-B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
4. **mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*]
5. **exit**
6. **ethernet cfm mep crosscheck start-delay** *delay*
7. **exit**
8. **ethernet cfm mep crosscheck** {**enable** | **disable**} **level** {*level-id* | *level-id-level-id* [,*level-id-level-id*]} **vlan** {*vlan-id* | **any** | *vlan-id-vlan-id* [,*vlan-id-vlan-id*]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain Customer level 7 direction outward`	Defines an outward CFM domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	**mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*] **Example:** `Router(config-ether-cfm)# mep crosscheck mpid 401 vlan 100`	Statically defines a remote MEP on a VLAN within a specified domain.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet cfm mep crosscheck start-delay** *delay* **Example:** `Router(config)# ethernet cfm mep crosscheck start-delay 60`	Configures the maximum amount of time that the device waits for remote MEPs to come up before the cross-check operation is started.
Step 7	**exit** **Example:** `Router(config)# exit`	Returns the CLI to privileged EXEC mode.
Step 8	**ethernet cfm mep crosscheck** {**enable** | **disable**} **level** {*level-id* | *level-id-level-id* [,*level-id-level-id*]} **vlan** {*vlan-id* | **any** | *vlan-id-vlan-id* [,*vlan-id-vlan-id*]} **Example:** `Router# ethernet cfm mep crosscheck enable level 7 vlan 100`	Enables cross-checking between MEPs.

Configuring CFM over Bridge Domains

Perform this task to configure Ethernet CFM over bridge domains. This task is optional.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id* **direction outward**
4. **service** *csi-id* **evc** *evc-name*
5. **exit**
6. **ethernet cfm domain** *domain-name* **level** *level-id*
7. **exit**
8. **ethernet cfm domain** *domain-name* **level** *level-id*
9. **service** *csi-id* **evc** *evc-name*
10. **mep crosscheck mpid** *id* **evc** *evc-name* **mac** *mac-address*
11. **exit**
12. **ethernet evc** *evc-name*
13. **exit**
14. **interface** *type number*
15. **no ip address**
16. **service instance** *id* **ethernet** *evc-id*
17. **encapsulation dot1q** *vlan-id*
18. **bridge-domain** *bridge-id*
19. **cfm mep domain** *domain-name* **outward mpid** *mpid-value*
20. **end**
21. **configure terminal**
22. **interface** *type name*
23. **no ip address**
24. **ethernet cfm mip level** *level-id*
25. **service instance** *id* **ethernet** *evc-id*
26. **encapsulation dot1q** *vlan-id*
27. **bridge-domain** *bridge-id*
28. **cfm mep domain** *domain-name* **inward mpid** *mpid-value*
29. **end**
30. **configure terminal**
31. **ethernet cfm cc enable level** *level-id* **evc** *evc-name*
32. **ethernet cfm cc level any evc** *evc-name* **interval** *seconds* **loss-threshold** *num-msgs*
33. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **direction outward** **Example:** `Router(config)# ethernet cfm domain CUSTOMER level 7 direction outward`	Defines a CFM maintenance domain at a particular level and enters Ethernet CFM configuration mode.
Step 4	**service** *csi-id* **evc** *evc-name* **Example:** `Router(config-ether-cfm)# service customer_100 evc evc_100`	Sets a universally unique ID for a CSI within a maintenance domain.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain MIP level 7`	Defines a CFM maintenance domain at a particular level and enters Ethernet CFM configuration mode.
Step 7	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 8	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config)# ethernet cfm domain PROVIDER level 4`	Defines a CFM maintenance domain at a particular level and enters Ethernet CFM configuration mode.
Step 9	**service** *csi-id* **evc** *evc-name* **Example:** `Router(config-ether-cfm)# service provider_1 evc evc_100`	Sets a universally unique ID for a CSI within a maintenance domain.
Step 10	**mep crosscheck mpid** *id* **evc** *evc-name* **mac** *mac-address* **Example:** `Router(config-ether-cfm)# mep crosscheck mpid 200 evc evc_100 mac 1010.1010.1010`	Statically defines a remote MEP within a maintenance domain.
Step 11	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 12	**ethernet evc** *evc-name* **Example:** `Router(config)# ethernet evc evc_100`	Defines an EVC and enters EVC configuration mode.
Step 13	**exit** **Example:** `Router(config-evc)# exit`	Returns the CLI to global configuration mode.
Step 14	**interface** *type number* **Example:** `Router(config)# interface Ethernet 1/0`	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 15	**no ip address** **Example:** `Router(config-if)# no ip address`	Disables IP processing.
Step 16	**service instance** *id* **ethernet** *evc-id* **Example:** `Router(config-if)# service instance 100 ethernet evc_100`	Specifies an Ethernet service instance on an interface and enters service instance configuration mode.
Step 17	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to map 802.1Q frames on an ingress interface to the appropriate service instance.
Step 18	**bridge-domain** *bridge-id* **Example:** `Router(config-if-srv)# bridge-domain 100`	Establishes a bridge domain.
Step 19	**cfm mep domain** *domain-name* **outward mpid** *mpid-value* **Example:** `Router(config-if-srv)# cfm mep domain CUSTOMER outward mpid 1001`	Configures a MEP for a domain.
Step 20	**end** **Example:** `Router(config-if-srv)# end`	Returns the CLI to privileged EXEC mode.
Step 21	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 22	**interface** *type name* **Example:** `Router(config)# interface Ethernet 1/1`	Specifies an interface and enters interface configuration mode.
Step 23	**no ip address** **Example:** `Router(config-if)# no ip address`	Disables IP processing.
Step 24	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 7`	Provisions a MIP at a specified maintenance level on an interface.
Step 25	**service instance** *id* **ethernet** *evc-id* **Example:** `Router(config-if)# service instance 100 ethernet evc_100`	Configures an Ethernet service instance on an interface and enters service instance configuration mode.
Step 26	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to map 802.1Q frames on an ingress interface to the appropriate service instance.
Step 27	**bridge-domain** *bridge-id* **Example:** `Router(config-if-srv)# bridge-domain 100`	Establishes a bridge domain.
Step 28	**cfm mep domain** *domain-name* **inward mpid** *mpid-value* **Example:** `Router(config-if-srv)# cfm mep domain PROVIDER inward mpid 201`	Configures a MEP for a domain.

	Command or Action	Purpose
Step 29	**end** **Example:** `Router(config-if-srv)# end`	Returns the CLI to privileged EXEC mode.
Step 30	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 31	**ethernet cfm cc enable level** *level-id* **evc** *evc-name* **Example:** `Router(config)# ethernet cfm cc enable level 0-7 evc evc_100`	Globally enables transmission of CCMs.
Step 32	**ethernet cfm cc level any evc** *evc-name* **interval** *seconds* **loss-threshold** *num-msgs* **Example:** `Router(config)# ethernet cfm cc level any evc evc_100 interval 100 loss-threshold 2`	Sets the parameters for CCMs.
Step 33	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Note When configuring CFM over bridge domains where the bridge-domain ID matches the vlan ID service, you must configure the vlan service and the EVC service with the same service name. The bridge-domain is associated with the EVC service. The vlan and the bridge-domain represent the same broadcast domain.

Troubleshooting Tips

To verify and isolate a fault, start at the highest level maintenance domain and do the following:

- Check the device error status.
- When a error exists, perform a loopback test to confirm the error.
- Run a traceroute to the destination to isolate the fault.
- If the fault is identified, correct the fault.

- If the fault is not identified, go to the next lower maintenance domain and repeat these four steps at that maintenance domain level.
- Repeat the first four steps, as needed, to identify and correct the fault.

Configuring Ethernet OAM Interaction with CFM

For Ethernet OAM to function with CFM, you must configure an EVC and the OAM manager and associate the EVC with CFM. Additionally, you must use an inward facing MEP when you want interaction with the OAM manager.

- Configuring the OAM Manager, page 235
- Enabling Ethernet OAM, page 237

Configuring the OAM Manager

Note If you configure, change, or remove a UNI service type, EVC, Ethernet service instance, or CE-VLAN configuration, all configurations are checked to ensure that UNI service types are matched with EVC configurations and Ethernet service instances are matched with CE-VLAN configurations. Configurations are rejected if the pairings do not match.

Perform this task to configure the OAM manager on a PE device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
4. **service** *csi-id* **vlan** *vlan-id*
5. **exit**
6. **ethernet evc** *evc-id*
7. **oam protocol** {**cfm svlan** *svlan-id* **domain** *domain-name* | **ldp**}
8. **exit**
9. Repeat Steps 3 through 8 to define other CFM domains that you want OAM manager to monitor.
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain cstmr1 level 3`	Defines a CFM domain, sets the domain level, and enters Ethernet CFM configuration mode.
Step 4	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service csi2 vlan 10`	Defines a universally unique customer service instance (CSI) and VLAN ID within the maintenance domain.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet evc** *evc-id* **Example:** `Router(config)# ethernet evc 50`	Defines an EVC and enters EVC configuration mode.
Step 7	**oam protocol** {**cfm svlan** *svlan-id* **domain** *domain-name* | **ldp**} **Example:** `Router(config-evc)# oam protocol cfm svlan 10 domain cstmr1`	Configures the EVC OAM protocol.

	Command or Action	Purpose

Step 8	**exit**
	
	Example:
	`Router(config-evc)# exit`
	
	Example:
	`Router(config)#`
		Returns the CLI to global configuration mode.

Step 9	Repeat Steps 3 through 8 to define other CFM domains that you want OAM manager to monitor.	--
Step 10	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Enabling Ethernet OAM

The order in which the global and interface configuration commands are issued determines the configuration. The last command that is issued has precedence.

Perform this task to enable Ethernet OAM on a device or on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet oam** [**max-rate** *oampdus* | **min-rate** *num-seconds*| **mode** {**active** | **passive**} | **timeout** *seconds*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1/3`	Specifies an interface and enters interface configuration mode.
Step 4	**ethernet oam** [**max-rate** *oampdus* | **min-rate** *num-seconds*| **mode** {**active** | **passive**} | **timeout** *seconds*] **Example:** `Router(config-if)# ethernet oam max-rate 50`	Enables Ethernet OAM on an interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Configuration Examples for Configuring Ethernet CFM in a Service Provider Network

- Example Provisioning a Network, page 238
- Example Provisioning Service, page 240

Example Provisioning a Network

This configuration example shows only CFM-related commands. All commands that are required to set up the data path and configure the VLANs on the device are not shown. However, it should be noted that CFM traffic will not flow into or out of the device if the VLANs are not properly configured.

```
CE-A
!
ethernet cfm domain Customer level 7 direction outward
!!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
```

```
!!
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
!
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
```

U-PE A

```
!
ethernet cfm domain Customer level 7
!
ethernet cfm domain ServiceProvider level 4
mep archive-hold-time 60
!
ethernet cfm domain OperatorA level 1
mep archive-hold-time 65
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
interface gigabitethernet4/2
ethernet cfm mip level 1
!
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
!
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
```

PE-AGG A

```
ethernet cfm domain OperatorA level 1
mep archive-hold-time 65
!
ethernet cfm enable
!
interface gigabitethernet3/1
ethernet cfm mip level 1
!
interface gigabitethernet4/1
ethernet cfm mip level 1
```

N-PE A

```
!
ethernet cfm domain ServiceProvider level 4
mep archive-hold-time 60
!
ethernet cfm domain OperatorA level 1
mep archive-hold-time 65
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
interface gigabitethernet3/0
ethernet cfm mip level 1
!
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
!
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
```

U-PE B

```
!
ethernet cfm domain Customer level 7
!
ethernet cfm domain ServiceProvider level 4
mep archive-hold-time 60
!
ethernet cfm domain OperatorB level 2
mep archive-hold-time 65
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
```

```
!
interface gigabitethernet2/0
ethernet cfm mip level 2
!
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
!
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
PE-AGG B
ethernet cfm domain OperatorB level 2
mep archive-hold-time 65
!
ethernet cfm enable
!
interface gigabitethernet1/1
ethernet cfm mip level 2
!
interface gigabitethernet2/1
ethernet cfm mip level 2
N-PE B
!
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
!
ethernet cfm domain ServiceProvider level 4
mep archive-hold-time 60
!
ethernet cfm domain OperatorB level 2
mep archive-hold-time 65
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
interface gigabitethernet1/2
ethernet cfm mip level 2
!
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
CE-B
!
ethernet cfm domain Customer level 7 direction outward
!!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!!
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
!
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
```

Example Provisioning Service

```
CE-A
!
ethernet cfm domain Customer level 7 direction outward
service Customer1 vlan 100
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
interface gigabitethernet3/2
ethernet cfm mep level 7 direction outward domain Customer1 mpid 701 vlan 100
!
ethernet cfm cc enable level 7 vlan 100
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
U-PE A
```

```
!
ethernet cfm domain Customer level 7
!
ethernet cfm domain ServiceProvider level 4
mep archive-hold-time 60
service MetroCustomer1 vlan 100
!
ethernet cfm domain OperatorA level 1
mep archive-hold-time 65
service MetroCustomer1OpA vlan 100
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
interface gigabitethernet3/2
ethernet cfm mip level 7
ethernet cfm mep level 4 mpid 401 vlan 100
ethernet cfm mep level 1 mpid 101 vlan 100
!
interface gigabitethernet4/2
ethernet cfm mip level 1
!
ethernet cfm cc enable level 4 vlan 100
ethernet cfm cc enable level 1 vlan 100
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
PE-AGG A
ethernet cfm domain OperatorA level 1
mep archive-hold-time 65
service MetroCustomer1OpA vlan 100
!
ethernet cfm enable
!
interface gigabitethernet3/1
ethernet cfm mip level 1
!
interface gigabitethernet4/1
ethernet cfm mip level 1
N-PE A
!
ethernet cfm domain ServiceProvider level 4
mep archive-hold-time 60
service MetroCustomer1 vlan 100
!
ethernet cfm domain OperatorA level 1
mep archive-hold-time 65
service MetroCustomer1OpA vlan 100
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
interface gigabitethernet3/0
ethernet cfm mip level 1
!
interface gigabitethernet4/0
ethernet cfm mip level 4
ethernet cfm mep level 1 mpid 102 vlan 100
!
ethernet cfm cc enable level 1 vlan 100
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
U-PE B
!
ethernet cfm domain Customer level 7
!
ethernet cfm domain ServiceProvider level 4
mep archive-hold-time 60
service MetroCustomer1 vlan 100
!
ethernet cfm domain OperatorB level 2
mep archive-hold-time 65
```

```
service MetroCustomer1OpB vlan 100
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
interface gigabitethernet1/0
ethernet cfm mip level 7
ethernet cfm mep level 4 mpid 402 vlan 100
ethernet cfm mep level 2 mpid 201 vlan 100
!
interface gigabitethernet2/0
ethernet cfm mip level 2
!
ethernet cfm cc enable level 4 vlan 100
ethernet cfm cc enable level 2 vlan 100
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
PE-AGG B
ethernet cfm domain OperatorB level 2
mep archive-hold-time 65
service MetroCustomer1OpB vlan 100
!
ethernet cfm enable
!
interface gigabitethernet1/1
ethernet cfm mip level 2
!
interface gigabitethernet2/1
ethernet cfm mip level 2
N-PE B
!
ethernet cfm domain ServiceProvider level 4
mep archive-hold-time 60
service MetroCustomer1 vlan 100
!
ethernet cfm domain OperatorB level 2
mep archive-hold-time 65
service MetroCustomer1OpB vlan 100
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
interface gigabitethernet1/2
ethernet cfm mip level 2
!
interface gigabitethernet2/2
ethernet cfm mip level 4
ethernet cfm mep level 2 mpid 202 vlan 100
!
ethernet cfm cc enable level 2 vlan 100
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
CE-B
!
ethernet cfm domain Customer level 7 direction outward
service Customer1 vlan 100
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
interface gigabitethernet3/2
ethernet cfm mep level 7 direction outward domain Customer1 mpid 702 vlan 100
!
ethernet cfm cc enable level 7 vlan 100
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
```

Additional References

Related Documents

Related Topic	Document Title
Carrier Ethernet commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Carrier Ethernet Command Reference*
Cisco IOS commands: master list of commands with complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS Master Command List, All Releases
Ethernet Local Management Interface on a provider edge device	Configuring Ethernet Local Management Interface at a Provider Edge
IP SLAs for Metro Ethernet	Configuring IP SLAs Metro-Ethernet 3.0 ITU T Y. 1731 Operations
IEEE 802.3ah	IEEE 802.3ah Ethernet in the First Mile
NSF/SSO and MPLS	NSF/SSO - MPLS LDP and LDP Graceful Restart
ISSU feature and functions	Cisco IOS Broadband High Availability In Service Software Upgrade
Performing an ISSU	Cisco IOS In Service Software Upgrade Process and Enhanced Fast Software Upgrade Process
SSO	"Stateful Switchover" chapter of the *Cisco IOS High Availability Configuration Guide*

Standards

Standard	Title
IEEE P802.1ag/D1.0	*Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks - Amendment 5: Connectivity Fault Management*
IETF VPLS OAM	*L2VPN OAM Requirements and Framework*
ITU-T	ITU-T Y.1731 OAM Mechanisms for Ethernet-Based Networks

MIBs

MIB	MIBs Link
CISCO-ETHER-CFM-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Ethernet CFM in a Service Provider Network

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 3 Feature Information for Configuring Ethernet Connectivity Fault Management in a Service Provider Network

Feature Name	Releases	Feature Information
CFM Outward Facing MEPs on Switch Ports	12.2(33)SRD Cisco IOS XE 3.1.0SG	The CFM Outward Facing MEPs on Switch Ports feature supports outward facing MEPs on switch ports. It is an enhancement to the Outward Facing MEP feature that supports the network at the distribution and access tiers. The following command was introduced or modified: **ethernet cfm mep level mpid vlan**.

Feature Name	Releases	Feature Information
Ethernet Connectivity Fault Management	12.2(33)SRA12.2(33)SRB 12.4(15)T2 12.2(33)SXI Cisco IOS XE 3.1.0SG	Ethernet CFM is an end-to-end per-service-instance Ethernet layer OAM protocol. It includes proactive connectivity monitoring, fault verification, and fault isolation for large Ethernet MANs and WANs. Ethernet CFM is supported on the Cisco 7600 router in Cisco IOS Release 12.2(33)SRA and on the Cisco 7200 VXR router in Cisco IOS Release 12.4(15)T. The following commands were introduced or modified: **clear ethernet cfm errors**, **clear ethernet cfm maintenance-points remote**, **clear ethernet cfm traceroute-cache**, **debug ethernet cfm all**, **debug ethernet cfm diagnostic**, **debug ethernet cfm errors**, **debug ethernet cfm events**, **debug ethernet cfm packets**, **ethernet cfm cc**, **ethernet cfm cc enable level vlan**, **ethernet cfm domain level**, **ethernet cfm enable**, **ethernet cfm enable (interface)**, **ethernet cfm mep crosscheck**, **ethernet cfm mep crosscheck start-delay**, **ethernet cfm mep level mpid vlan**, **ethernet cfm mip level**, **ethernet cfm traceroute cache**, **ethernet cfm traceroute cache hold-time**, **ethernet cfm traceroute cache size**, **mep archive-hold-time**, **ping ethernet mpid vlan**, **ping ethernet vlan**, **service vlan**, **show ethernet cfm errors**, **show ethernet cfm maintenance-points local**, **show ethernet cfm maintenance-points remote**, **show ethernet cfm maintenance-points remote crosscheck**, **show ethernet cfm maintenance-points remote detail**, **show ethernet cfm traceroute-cache**, **snmp-server**

Feature Name	Releases	Feature Information
		enable traps ethernet cfm cc, **snmp-server enable traps ethernet cfm crosscheck**, **traceroute ethernet vlan**.
802.3ah and CFM Interworking	12.2(33)SRB 12.2(33)SXI Cisco IOS XE 3.1.0SG	The Ethernet OAM and Ethernet CFM Interworking feature enables Ethernet OAM and CFM to function together in a network.
Ethernet-OAM3.0: CFM Over BD, Untagged	12.2(33)SRD 12.2(50)SY	Ethernet-OAM3.0 with support for CFM over bridge domains is supported on the Cisco 7600 Series Route Switch Processor 720 and on the Cisco 7600 Series Supervisor Engine 720 in Cisco IOS Release 12.2(33)SRD. The following commands were introduced or modified: **cfm encapsulation**, **cfm mep domain**, **debug ethernet cfm all**, **debug ethernet cfm events**, **debug ethernet cfm packets**, **ethernet cfm cc**, **ethernet cfm cc enable level evc**, **ethernet cfm mep crosscheck**, **mep crosscheck mpid evc**, **mep crosscheck mpid vlan, ping ethernet evc, service evc**, **show ethernet cfm maintenance-points remote crosscheck**, **show ethernet cfm maintenance-points remote detail**, **traceroute ethernet evc**.
ISSU Support in CFM 802.1ag/ 1.0d	12.2(33)SRD	ISSU support allows a Cisco IOS software product to perform and upgrade or downgrade without disrupting packet flow. The following command was introduced or modified: **debug ethernet cfm ha**.
NSF/SSO Support in CFM 802.1ag/1.0d	12.2(33)SRD Cisco IOS XE 3.1.0SG	CFM support for NSF/SSO allows CFM processes that support dual route processors in active/standby mode to continue forwarding packets following a switchover.

Feature Name	Releases	Feature Information
Outward Facing MEP	12.4(11)T 12.2(33)SRB 12.2(33)SXI	The Outward Facing MEP feature is an enhancement to Ethernet CFM that supports the distribution and access environments by supporting outward facing MEPs on routed (Layer 3) ports. Ethernet CFM with support for outward facing MEPs is supported on the Cisco Integrated Services Routers (ISRs) in Cisco IOS Release 12.4(11)T. The following command was introduced or modified: **ethernet cfm mep level mpid vlan**.

Glossary

CCM --continuity check message. A multicast CFM frame that a MEP transmits periodically to ensure continuity across the maintenance entities to which the transmitting MEP belongs, at the MA level on which the CCM is sent. No reply is sent in response to receiving a CCM.

EVC --Ethernet virtual connection. An association of two or more user-network interfaces.

fault alarm --An out-of-band signal, typically an SNMP notification, that notifies a system administrator of a connectivity failure.

inward-facing MEP --A MEP that resides in a bridge and transmits to and receives CFM messages from the direction of the bridge relay entity.

maintenance domain --The network or part of the network belonging to a single administration for which faults in connectivity are to be managed. The boundary of a maintenance domain is defined by a set of DSAPs, each of which may become a point of connectivity to a service instance.

maintenance domain name --The unique identifier of a domain that CFM is to protect against accidental concatenation of service instances.

MEP --maintenance endpoint. An actively managed CFM entity associated with a specific DSAP of a service instance, which can generate and receive CFM frames and track any responses. It is an endpoint of a single MA, and terminates a separate maintenance entity for each of the other MEPs in the same MA.

MEP CCDB --A database, maintained by every MEP, that maintains received information about other MEPs in the maintenance domain.

MIP --maintenance intermediate point. A CFM entity, associated with a specific pair of ISS SAPs or EISS Service Access Points, which reacts and responds to CFM frames. It is associated with a single maintenance association and is an intermediate point within one or more maintenance entities.

MIP CCDB --A database of information about the MEPs in the maintenance domain. The MIP CCDB can be maintained by a MIP.

MP --maintenance point. Either a MEP or a MIP.

MPID --maintenance endpoint identifier. A small integer, unique over a given MA, that identifies a specific MEP.

OAM --operations, administration, and maintenance. A term used by several standards bodies to describe protocols and procedures for operating, administrating, and maintaining networks. Examples are ATM OAM and IEEE Std. 802.3ah OAM.

operator --Entity that provides a service provider a single network of provider bridges or a single Layer 2 or Layer 3 backbone network. An operator may be identical to or a part of the same organization as the service provider. For purposes of IEEE P802.1ag, Draft Standard for Local and Metropolitan Area Networks, the operator and service provider are presumed to be separate organizations.

Terms such as "customer," "service provider," and "operator" reflect common business relationships among organizations and individuals that use equipment implemented in accordance with IEEE P802.1ag.

UNI --user-network interface. A common term for the connection point between an operator's bridge and customer equipment. A UNI often includes a C-VLAN-aware bridge component. The term UNI is used broadly in the IEEE P802.1ag standard when the purpose for various features of CFM are explained. UNI has no normative meaning.

Configuring Ethernet Local Management Interface at a Provider Edge

The advent of Ethernet as a metropolitan-area network (MAN) and WAN technology imposes a new set of operations, administration, and management (OAM) requirements on Ethernet's traditional operations, which had been centered on enterprise networks only. The expansion of Ethernet technology into the domain of service providers, where networks are substantially larger and more complex than enterprise networks and the user-base is wider, makes operational management of link uptime crucial. More importantly, the timeliness in isolating and responding to a failure becomes mandatory for normal day-to-day operations, and OAM translates directly to the competitiveness of the service provider.

This module provides general information about configuring Ethernet Local Management Interface (LMI), an OAM protocol, on a provider edge (PE) device.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring Ethernet Local Management Interface at a Provider Edge

- Ethernet OAM must be operational in the network.
- For Ethernet OAM to operate, the PE side of a connection must be running Ethernet Connectivity Fault Management (CFM) and Ethernet LMI.
- All VLANs used on a PE device to connect to a customer edge (CE) device must also be created on that CE device.
- To use Non-Stop Forwarding (NSF) and In Service Software Upgrade (ISSU), Stateful Switchover (SSO) must be configured and working properly.

Restrictions for Configuring Ethernet Local Management Interface at a Provider Edge

- Ethernet LMI is not supported on routed ports, EtherChannel port channels, or ports that belong to an EtherChannel, private VLAN ports, IEEE 802.1Q tunnel ports, or Ethernet over Multiprotocol Label Switching (MPLS) ports.
- Ethernet LMI cannot be configured on VLAN interfaces.
- The High Availability (HA) features NSF/SSO--E-LMI Support and ISSU--E-LMI Support are not supported on a CE device.

Information About Configuring Ethernet Local Management Interface at a Provider Edge

Ethernet Virtual Circuit

An Ethernet virtual circuit (EVC) as defined by the Metro Ethernet Forum is a port level point-to-point or multipoint-to-multipoint Layer 2 circuit. EVC status can be used by a CE device to find an alternative path in to the service provider network or in some cases, fall back to a backup path over Ethernet or another alternative service such as Frame Relay or ATM.

Ethernet LMI

Ethernet LMI is an Ethernet OAM protocol between a CE device and a PE device. Ethernet LMI provides CE devices with the status of EVCs for large Ethernet MANs and WANs and provides information that enables CE devices to autoconfigure. Specifically, Ethernet LMI runs on the PE-CE user network interface

(UNI) link and notifies a CE device of the operating state of an EVC and the time when an EVC is added or deleted. Ethernet LMI also communicates the attributes of an EVC.

Ethernet LMI interoperates with Ethernet CFM, an OAM protocol that runs within the provider network to collect OAM status. Ethernet CFM runs at the provider maintenance level (user provider edge [UPE] to UPE at the UNI). Ethernet LMI relies on the OAM Ethernet Infrastructure (EI) to interwork with CFM to learn the end-to-end status of EVCs across CFM domains.

Ethernet LMI is disabled globally by default. When Ethernet LMI is enabled globally, all interfaces are automatically enabled. Ethernet LMI can also be enabled or disabled at the interface to override the global configuration. The last Ethernet LMI command issued is the command that has precedence. No EVCs, Ethernet service instances, or UNIs are defined, and the UNI bundling service is bundling with multiplexing.

Ethernet CFM

Ethernet CFM is an end-to-end per-service-instance (per VLAN) Ethernet layer OAM protocol that includes proactive connectivity monitoring, fault verification, and fault isolation. End-to-end CFM can be from PE device to PE device or from CE device to CE device. For more information about Ethernet CFM, see “ Configuring Ethernet Connectivity Fault Management in a Service Provider Network” in the *Cisco IOS Carrier Ethernet Configuration Guide* .

OAM Manager

The OAM manager is an infrastructure element that streamlines interaction between OAM protocols. The OAM manager requires two interworking OAM protocols, in this case, Ethernet CFM and Ethernet LMI. No interactions are required between Ethernet LMI and the OAM manager on the CE side. On the UPE side, the OAM manager defines an abstraction layer that relays data collected from Ethernet CFM to the Ethernet LMI device.

Ethernet LMI and OAM manager interaction is unidirectional, from the OAM manager to Ethernet LMI on the UPE side of the device. An information exchange results from an Ethernet LMI request or is triggered by the OAM manager when it receives notification from the OAM protocol that the number of UNIs has changed. A change in the number of UNIs may cause a change in EVC status.

The OAM manager calculates EVC status given the number of active UNIs and the total number of associated UNIs. You must configure CFM to notify the OAM manager of all changes to the number of active UNIs or to the remote UNI ID for a given service provider VLAN (S-VLAN) domain.

The information exchanged includes the following:

- EVC name and availability status (active, inactive, partially active, or not defined)
- Remote UNI name and status (up, disconnected, administratively down, excessive frame check sequence [FCS] failures, or not reachable)
- Remote UNI counts (the total number of expected UNIs and the number of active UNIs)

Benefits of Ethernet LMI at a Provider Edge

- Communication of end-to-end status of the EVC to the CE device
- Communication of EVC and UNI attributes to a CE device
- Competitive advantage for service providers

HA Features Supported by Ethernet LMI

In access and service provider networks using Ethernet technology, HA is a requirement, especially on Ethernet OAM components that manage EVC connectivity. End-to-end connectivity status information is critical and must be maintained on a hot standby route processor (RP) (a standby RP that has the same software image as the active RP and supports synchronization of line card, protocol, and application state information between RPs for supported features and protocols).

End-to-end connectivity status is maintained on the CE, PE, and access aggregation PE (uPE) network nodes based on information received by protocols such as Ethernet LMI, CFM, and 802.3ah. This status information is used to either stop traffic or switch to backup paths when an EVC is down.

The NSF/SSO and ISSU support enhancements are introduced and enabled automatically during configuration of the Cisco 7600 router.

Metro Ethernet clients (E-LMI, CFM, 802.3ah) maintain configuration data and dynamic data, which is learned through protocols. Every transaction involves either accessing or updating data in the various databases. If the database is synchronized across active and standby modules, the modules are transparent to clients.

The Cisco IOS infrastructure provides component Application Programming Interfaces (APIs) that are helpful in maintaining a hot standby RP. Metro Ethernet HA clients (E-LMI, HA/ISSU, CFM HA/ISSU, 802.3ah HA/ISSU) interact with these components, update the database, and trigger necessary events to other components.

- Benefits of Ethernet LMI HA, page 254

Benefits of Ethernet LMI HA

- Elimination of network downtime for Cisco IOS software image upgrades, resulting in higher availability.
- Elimination of resource scheduling challenges associated with planned outages and late night maintenance windows
- Accelerated deployment of new services and applications and faster implementation of new features, hardware, and fixes due to the elimination of network downtime during upgrades.
- Reduced operating costs due to outages while delivering higher service levels due to the elimination of network downtime during upgrades

NSF SSO Support in E-LMI

The redundancy configurations SSO and NSF are supported in Ethernet LMI and are automatically enabled. A switchover from an active to a standby RP occurs when the active RP fails, is removed from the networking device, or is manually taken down for maintenance. The primary function of Cisco NSF is to continue forwarding IP packets following an RP switchover. NSF also interoperates with the SSO feature to minimize network downtime following a switchover.

For detailed information about the SSO feature, see the ‘Stateful Switchover’ chapter of the *Cisco IOS High Availability Configuration Guide*. For detailed information about the NSF feature, see the ‘Cisco Nonstop Forwarding’ chapter of the *Cisco IOS High Availability Configuration Guide*.

ISSU Support in E-LMI

ISSU allows you to perform a Cisco IOS software upgrade or downgrade without disrupting packet flow. E-LMI performs updates of the parameters within the Ethernet LMI database to the standby RP. This checkpoint data requires ISSU capability to transform messages from one release to another. All the components that perform active RP to standby RP updates using messages require ISSU support. ISSU is automatically enabled in Ethernet LMI.

ISSU lowers the impact that planned maintenance activities have on network availability by allowing software changes while the system is in service. For detailed information about ISSU, see the Cisco OS In Service Software Upgrade Process chapter of the *Cisco IOS High Availability Configuration Guide*.

How to Configure Ethernet Local Management Interface at a Provider Edge

Configuring Ethernet LMI Interaction with CFM

For Ethernet LMI to function with CFM, you must configure EVCs, Ethernet service instances, and Ethernet LMI customer VLAN mapping. Most of the configuration occurs on the PE device on the interfaces connected to the CE. On the CE device, you need only enable Ethernet LMI on the connecting interface. Also, you must configure some OAM parameters; for example, EVC definitions on PE devices on both sides of a metro network.

CFM and OAM interworking requires an inward facing Maintenance Entity Group End Point (MEP).

Configuring the OAM Manager

Note If you configure, change, or remove a UNI service type, EVC, Ethernet service instance, or CE-VLAN configuration, all configurations are checked to ensure that the configurations match (UNI service type with EVC or Ethernet service instance and CE-VLAN configuration). The configuration is rejected if the configurations do not match.

Perform this task to configure the OAM manager on a PE device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
4. **service** *csi-id* **vlan** *vlan-id*
5. **exit**
6. **ethernet evc** *evc-id*
7. **oam protocol** {**cfm svlan** *svlan-id* **domain** *domain-name* | **ldp**}
8. **uni count** *value* [**multipoint**]
9. **exit**
10. Repeat Steps 3 through 9 to define other CFM domains that you want OAM manager to monitor.
11. **interface** *type number*
12. **service instance** *id* **ethernet** [*evc-id*]
13. **ethernet lmi ce-vlan map** {*vlan-id* [**untagged**] | **any** | **default** | **untagged**}
14. **exit**
15. **ethernet uni** [**bundle** [**all-to-one**] | **id** *uni-id*| **multiplex**]
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain cstmr1 level 3`	Defines a CFM domain, sets the domain level and places the command-line interface (CLI) in Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 4	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service csi2 vlan 10`	Defines a universally unique customer service instance (CSI) and VLAN ID within the maintenance domain.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet evc** *evc-id* **Example:** `Router(config)# ethernet evc 50`	Defines an EVC and enters EVC configuration mode.
Step 7	**oam protocol** {**cfm svlan** *svlan-id* **domain** *domain-name* | **ldp**} **Example:** `Router(config-evc)# oam protocol cfm svlan 10 domain cstmr1`	Configures the EVC OAM protocol as CFM and identifies the S-VLAN-ID for the CFM domain maintenance level as configured in Steps 3 and 4. **Note** If the CFM domain does not exist, this command is rejected, and an error message is displayed.
Step 8	**uni count** *value* [**multipoint**] **Example:** `Router(config-evc)# uni count 3`	(Optional) Sets the UNI count for the EVC. • If this command is not issued, the service defaults to a point-to-point service. If a value of 2 is entered, point-to-multipoint service becomes an option. If a value of 3 or greater is entered, the service is point-to-multipoint. **Note** If you enter a number greater than the number of endpoints, the UNI status is partially active even if all endpoints are up. If you enter a UNI count less than the number of endpoints, status might be active, even if all endpoints are not up.
Step 9	**exit** **Example:** `Router(config-evc)# exit`	Returns the CLI to global configuration mode.
Step 10	Repeat Steps 3 through 9 to define other CFM domains that you want OAM manager to monitor.	

	Command or Action	Purpose
Step 11	**interface** *type number* **Example:** `Router(config)# interface ethernet 1/3`	Specifies a physical interface connected to the CE device and enters interface configuration mode.
Step 12	**service instance** *id* **ethernet** [*evc-id*] **Example:** `Router(config-if)# service instance 400 ethernet 50`	Configures an Ethernet service instance on the interface and enters Ethernet service configuration mode. • The Ethernet service instance identifier is a per-interface service identifier and does not map to a VLAN.
Step 13	**ethernet lmi ce-vlan map** {*vlan-id* [**untagged**] | **any** | **default** | **untagged**} **Example:** `Router(config-if-srv)# ethernet lmi ce-vlan map 30`	Configures an Ethernet LMI customer VLAN-to-EVC map for a particular UNI.
Step 14	**exit** **Example:** `Router(config-if-srv)# exit`	Returns the CLI to interface configuration mode.
Step 15	**ethernet uni** [**bundle** [**all-to-one**] | **id** *uni-id*| **multiplex**] **Example:** `Router(config-if)# ethernet uni bundle`	Sets UNI bundling attributes.
Step 16	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Enabling Ethernet LMI

The order in which the global and interface configuration commands are issued determines the configuration. The last command that is issued has precedence.

Perform this task to enable Ethernet LMI on a device or on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet lmi interface**
5. **ethernet lmi** {**n393** *value* | **t392** *value*}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 1/3`	Defines an interface to configure as an Ethernet LMI interface and enters interface configuration mode.
Step 4	**ethernet lmi interface** **Example:** `Router(config-if)# ethernet lmi interface`	Configures Ethernet LMI on the interface. When Ethernet LMI is enabled globally, it is enabled on all interfaces unless you disable it on specific interfaces. If Ethernet LMI is disabled globally, you can use this command to enable it on specified interfaces.
Step 5	**ethernet lmi** {**n393** *value* | **t392** *value*} **Example:** `Router(config-if)# ethernet lmi n393 10`	Configures Ethernet LMI parameters for the UNI.

	Command or Action	Purpose
Step 6	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Displaying Ethernet LMI and OAM Manager Information

Perform this task to display Ethernet LMI or OAM manager information. All the steps are optional and can be performed in any order.

SUMMARY STEPS

1. **enable**
2. **show ethernet lmi** {{**evc** [**detail** *evc-id* [**interface** *type number*] | **map interface** *type number*]} | {**parameters** | **statistics**} **interface** *type number* | **uni map** [**interface** *type number*]}
3. **show ethernet service evc** [**detail** | **id** *evc-id* [**detail**] | **interface** *type number*[**detail**]]
4. **show ethernet service instance** [**detail** | **id** *id* | **interface** *type number* | **policy-map** | **stats**]
5. **show ethernet service interface** [*type number*] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ethernet lmi** {{**evc** [**detail** *evc-id* [**interface** *type number*] | **map interface** *type number*]} | {**parameters** | **statistics**} **interface** *type number* | **uni map** [**interface** *type number*]} **Example:** `Router# show ethernet lmi evc`	Displays information that was sent to the CE.
Step 3	**show ethernet service evc** [**detail** | **id** *evc-id* [**detail**] | **interface** *type number*[**detail**]] **Example:** `Router# show ethernet service evc`	Displays information about all EVCs or about a specified EVC.

	Command or Action	Purpose
Step 4	**show ethernet service instance** [**detail** | **id** *id* | **interface** *type number* | **policy-map** | **stats**] **Example:** `Router# show ethernet service instance detail`	Displays information about customer service instances. • This example shows detailed information about all service instances (see the following section).
Step 5	**show ethernet service interface** [*type number*] [**detail**] **Example:** `Router# show ethernet service interface ethernet 1/3 detail`	Displays interface-only information about Ethernet customer service instances for all interfaces or for a specified interface. • This example shows detailed information about service instances for interface Ethernet 1/3 (see the following section).

Examples

The following example shows sample output from the **show ethernet lmi**command using the **evc** keyword:

```
Router# show ethernet lmi evc
St  EVC Id                                                         Port
--- ------------------------------------------------------------ --------------
A  EVC_MP2MP_101                                                  Gi0/1
A  EVC_P2P_110                                                    Gi0/1
```

The following example shows sample output from the **show ethernet service evc** command:

```
Router# show ethernet service evc
Identifier                    Type  Act-UNI-cnt Status
50                            MP-MP     0       NotDefined
```

The following example shows sample output from the **show ethernet service interface** command using the **detail** keyword:

```
Router# show ethernet service interface ethernet 1/3 detail
Interface: Ethernet1/3
ID: uni2
CE-VLANS: 30
EVC Map Type: Bundling
Associated EVCs:
    EVC-ID                        CE-VLAN
    50                            30
Associated Service Instances:
    Service-Instance-ID CE-VLAN
    400                 30
```

The following example shows sample output from the **show ethernet service instance**command using the **detail** keyword:

```
Router# show ethernet service instance detail

Service Instance ID: 400
Associated Interface: Ethernet1/3
Associated EVC: 50
CE-Vlans: 30
State: AdminDown
EFP Statistics:
   Pkts In   Bytes In   Pkts Out  Bytes Out
         0          0          0          0
```

Configuration Examples for Ethernet Local Management Interface at a Provider Edge

Example Ethernet OAM Manager on a PE Device Configuration

This example shows a sample configuration of OAM manager, CFM, and Ethernet LMI on a PE device:

```
Router# configure terminal
Router(config)# ethernet cfm domain Top level 7
Router(config)# ethernet cfm domain Provider level 4
Router(config-ether-cfm)# service customer_1 vlan 101
Router(config-ether-cfm)# mep crosscheck mpid 404 vlan 101
Router(config-ether-cfm)# exit
Router(config)# ethernet cfm domain Operator_level 2
Router(config-ether-cfm)# service operator_1 vlan 101
Router(config-ether-cfm)# exit
Router(config)# ethernet cfm enable
Router(config)# ethernet evc test1
Router(config-evc)# oam protocol cfm svlan 101 domain Provider
Router(config-evc)# exit
Router(config)# ethernet evc 101
Router(config-evc)# uni count 3
Router(config-evc)# oam protocol cfm svlan 101 domain Operator
Router(config-evc)# exit
Router(config)# ethernet lmi global
Router(config)# interface gigabitethernet 1/0/2
Router(config-if)# service instance 101 ethernet test1
Router(config-if-srv)# ethernet lmi ce-vlan map 101
Router(config-if-srv)# exit
Router(config-if)# exit
Router(config)# ethernet cfm cc enable level 2-4 vlan 101

Router(config)# exit
```

Example Ethernet OAM Manager on a CE Device Configuration

This example shows how to configure Ethernet LMI globally on a CE device:

```
Router# configure terminal
Router(config)# ethernet lmi global
Router(config)# exit
```

Additional References

Related Documents

Related Topic	Document Title
Ethernet CFM	"Configuring Ethernet Connectivity Fault Management in a Service Provider Network" in the *Cisco IOS Carrier Ethernet Configuration Guide*
Ethernet LMI	"Enabling Ethernet Local Management Interface" in the *Cisco IOS Carrier Ethernet Configuration Guide*
Remote Port Shutdown feature	"Configuring Remote Port Shutdown" in the *Cisco IOS Carrier Ethernet Configuration Guide*
IEEE 802.3ah	*IEEE 802.3ah Ethernet in the First Mile*
Cisco IOS HA configuration information	*Cisco IOS High Availability Configuration Guide*
Ethernet LMI commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Carrier Ethernet Command Reference*
Cisco IOS commands: master list of commands with complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS Master Command List, All Releases

Standards

Standard	Title
IEEE P802.1ag/D5.2	*Draft Standard for Local and Metropolitan Area Networks*
ITU-T	*ITU-T Y.1731 OAM Mechanisms for Ethernet-Based Networks*
IETF VPLS OAM	*L2VPN OAM Requirements and Framework*
Metro Ethernet Forum 16 Technical Specification	*Technical Specification MEF 16- Ethernet Local Management Interface*
ITU-T Q.3/13	*Liaison statement on Ethernet OAM (Y.17ethoam)*

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Ethernet Local Management Interface at a Provider Edge

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 4 Feature Information for Configuring Ethernet Local Management Interface at a Provider Edge

Feature Name	Releases	Feature Information
Ethernet Local Management Interface at a Provider Edge	12.2(33)SRB 12.2(33)SXI	Ethernet LMI is an Ethernet OAM protocol between a CE device and a PE device. Ethernet LMI provides CE devices with the status of EVCs for large Ethernet MANs and WANs and provides information that enables CE devices to autoconfigure. Specifically, Ethernet LMI runs on the PE-CE UNI link and notifies a CE device of the operating state of an EVC and when an EVC is added or deleted. Ethernet LMI also communicates the attributes of an EVC. In Cisco IOS Release 12.2(33)SRB, this feature was introduced on the Cisco 7600 series router. The following commands were introduced or modified: **debug ethernet lmi**, **debug ethernet service, ethernet evc**, **ethernet lmi ce-vlan map**, **ethernet uni**, **oam protocol**, **service instance ethernet**, **show ethernet service evc**, **show ethernet service instance**, **show ethernet service interface, uni count**.
ISSU Support in E-LMI	12.2(33)SRD 15.0(1)S	ISSU allows you to perform a Cisco IOS software upgrade or downgrade without disrupting packet flow. ISSU lowers the impact that planned maintenance activities have on network availability by allowing software changes while the system is in service. In Cisco IOS Release 12.2(33)SRD, this feature was introduced on the Cisco 7600 series router. The following commands were introduced or modified: **debug ethernet lmi**.

Feature Name	Releases	Feature Information
NSF/SSO Support in E-LMI	12.2(33)SRD 15.0(1)S	The redundancy configurations SSO and NSF are supported in Ethernet LMI and are automatically enabled. A switchover from an active to a standby RP occurs when the active RP fails, is removed from the networking device, or is manually taken down for maintenance. The primary function of Cisco NSF is to continue forwarding IP packets following an RP switchover. NSF also interoperates with the SSO feature to minimize network downtime following a switchover. In Cisco IOS Release 12.2(33)SRD, this feature was introduced on the Cisco 7600 series router. The following commands were introduced or modified: **debug ethernet lmi**.

Using Link Layer Discovery Protocol in Multivendor Networks

Link Layer Discovery Protocol (LLDP), standardized by the IEEE as part of 802.1ab, enables standardized discovery of nodes, which in turn facilitates future applications of standard management tools such as Simple Network Management Protocol (SNMP) in multivendor networks. Using standard management tools makes physical topology information available and helps network administrators detect and correct network malfunctions and inconsistencies in configuration.

Media Endpoint Discovery (MED) is an LLDP enhancement that was formalized by the Telecommunications Industry Association (TIA) for voice over IP (VoIP) applications.

The Cisco implementation of LLDP is based on the IEEE 802.1ab standard. This document describes LLDP and LLDP-MED and how they are supported in Cisco IOS software.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Using Link Layer Discovery Protocol in Multivendor Networks

- Type-Length-Value (TLV) types 0 through 127

- To support LLDP-MED, the following organizationally specific TLVs must be implemented:
  - Extended Power-via-Media Dependent Interface (MDI)
  - Inventory
  - LLDP-MED Capabilities
  - MAC/PHY Configuration Status
  - Network Policy
  - Port VLAN ID

Restrictions for Using Link Layer Discovery Protocol in Multivendor Networks

- Use of LLDP is limited to 802.1 media types such as Ethernet, Token Ring, and Fiber Distributed Data Interface (FDDI) networks.
- The maximum number of neighbor entries per chassis is limited on MED-capable network connectivity devices.

Information About Using Link Layer Discovery Protocol in Multivendor Networks

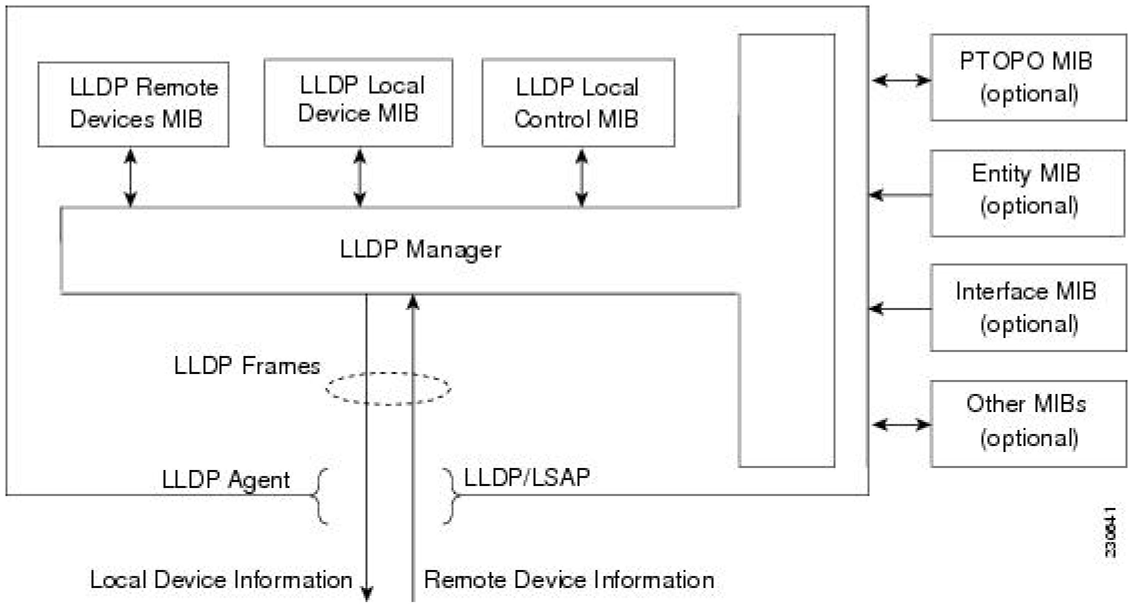
IEEE 802.1ab LLDP

IEEE 802.1ab LLDP is an optional link layer protocol for network topology discovery in multivendor networks. Discovery information includes device identifiers, port identifiers, versions, and other details. As a protocol that aids network management, LLDP provides accurate network mapping, inventory data, and network troubleshooting information.

LLDP is unidirectional, operating only in an advertising mode. LLDP does not solicit information or monitor state changes between LLDP nodes. LLDP periodically sends advertisements to a constrained multicast address. Devices supporting LLDP can send information about themselves while they receive and record information about their neighbors. Additionally, devices can choose to turn off the send or receive functions independently. Advertisements are sent out and received on every active and enabled interface, allowing any device in a network to learn about all devices to which it is connected. Applications that use this information include network topology discovery, inventory management, emergency services, VLAN assignment, and inline power supply.

Note LLDP and Cisco Discovery Protocol (CDP) can operate on the same interface.

The figure below shows a high-level view of LLDP operating in a network node.



When you configure LLDP or CDP location information on a per-port basis, remote devices can send Cisco medianet location information to the switch. For more information, refer to the Using Cisco Discovery Protocol module.

CDP allows only one wired switch to report the location information. CDP identifies and indicates the location of neighbor ports and ensures not to report the duplicate location information of the same device.

LLDP-MED

LLDP-MED operates between several classes of network equipment such as IP phones, conference bridges, and network connectivity devices such as routers and switches. By default, a network connectivity device sends out only LLDP packets until it receives LLDP-MED packets from an endpoint device. The network device then sends out LLDP-MED packets until the remote device to which it is connected ceases to be LLDP-MED capable.

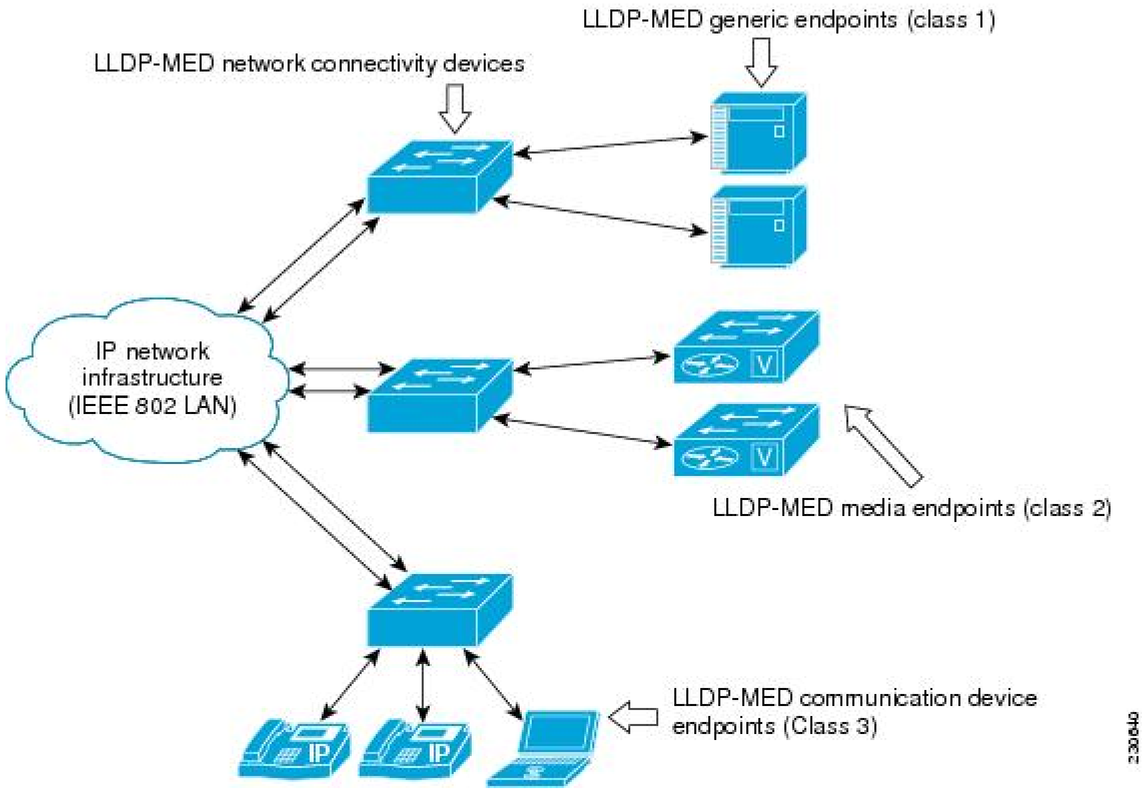
- Classes of Endpoints, page 269
- Types of Discovery Supported, page 270
- Benefits of LLDP-MED, page 270

Classes of Endpoints

LLDP-MED network connectivity devices provide IEEE 802 network access to LLDP-MED endpoints. LLDP-MED supports the following three classes of endpoints:

- Generic (class 1)--Basic participant endpoints; for example, IP communications controllers.
- Media (class 2)--Endpoints that support media streams; for example, media gateways and conference bridges.
- Communication Device (class 3)--Endpoints that support IP communications end users; for example, IP phones and Softphone.

The figure below shows an LLDP-MED-enabled LAN.



Types of Discovery Supported

LLDP-MED provides support to discover the following types of information, which are crucial to efficient operation and management of endpoint devices and the network devices supporting them:

- **Capabilities** --Endpoints determine the types of capabilities that a connected device supports and which ones are enabled.
- **Inventory** --LLDP-MED support exchange of hardware, software, and firmware versions, among other inventory details.
- **LAN speed and duplex** --Devices discover mismatches in speed and duplex settings.
- **Location identification** --An endpoint, particularly a telephone, learns its location from a network device. This location information may be used for location-based applications on the telephone and is important when emergency calls are placed.
- **Network policy** --Network connectivity devices notify telephones about the VLANs they should use.
- **Power** --Network connectivity devices and endpoints exchange power information. LLDP-MED provides information about how much power a device needs and how a device is powered. LLDP-MED also determines the priority of the device for receiving power.

Benefits of LLDP-MED

- Follows an open standard
- Supports E-911 emergency service, which is aided by location management
- Provides fast start capability
- Supports interoperability between multivendor devices

- Supports inventory management (location, version, etc.)
- Provides MIB support
- Supports plug and play installation
- Provides several troubleshooting (duplex, speed, network policy) mechanisms

TLV Elements

LLDP and LLDP-MED use TLVs to exchange information between network and endpoint devices. TLV elements are embedded in communications protocol advertisements and used for encoding optional information. The size of the type and length fields is fixed at 2 bytes. The size of the value field is variable. The type is a numeric code that indicates the type of field that this part of the message represents, and the length is the size of the value field, in bytes. The value field contains the data for this part of the message.

LLDP-MED supports the following TLVs:

- LLDP-MED capabilities TLV--Allows LLDP-MED endpoints to determine the capabilities that the connected device supports and has enabled.
- Network policy TLV--Allows both network connectivity devices and endpoints to advertise VLAN configurations and associated Layer 2 and Layer 3 attributes for the specific application on that port. For example, the switch can notify a phone of the VLAN number that it should use. The phone can connect to any switch, obtain its VLAN number, and then start communicating with the call control.

By defining a network-policy profile TLV, you can create a profile for voice and voice signalling by specifying the values for VLAN, class of service (CoS), differentiated services code point (DSCP), and tagging mode. These profile attributes are then maintained centrally on the switch and propagated to the phone.

- Power management TLV--Enables advanced power management between LLDP-MED endpoint and network connectivity devices. Allows switches and phones to convey power information, such as how the device is powered, power priority, and how much power the device needs. Supports advertisement of fractional wattage power requirements, endpoint power priority, and endpoint and network connectivity-device power status but does not provide for power negotiation between the endpoint and the network connectivity devices. When LLDP is enabled and power is applied to a port, the power TLV determines the actual power requirement of the endpoint device so that the system power budget can be adjusted accordingly. The switch processes the requests and either grants or denies power based on the current power budget. If the request is granted, the switch updates the power budget. If the request is denied, the switch turns off power to the port, generates a syslog message, and updates the power budget. If LLDP-MED is disabled or if the endpoint does not support the LLDP-MED power TLV, the initial allocation value is used throughout the duration of the connection.

Note A system power budget is the default power allocated to a device based on its device class. However, the total power that can be sourced from a switch is finite, and there will be some power budgeting done by the power module based on the number of ports already being served, total power that can be served, and how much new ports are requesting.

- Inventory management TLV--Allows an endpoint to send detailed inventory information about itself to the switch, including information hardware revision, firmware version, software version, serial number, manufacturer name, model name, and asset ID TLV.
- Location TLV--Provides location information from the switch to the endpoint device. The location TLV can send this information:

- Civic location information--Provides the civic address information and postal information. Examples of civic location information are street address, road name, and postal community name information.
- ELIN location information--Provides the location information of a caller. The location is determined by the Emergency location identifier number (ELIN), which is a phone number that routes an emergency call to the local public safety answering point (PSAP) and which the PSAP can use to call back the emergency caller.

Benefits of LLDP

- Follows IEEE 802.1ab standard.
- Enables interoperability among multivendor devices.
- Facilitates troubleshooting of enterprise networks and uses standard network management tools.
- Provides extension for applications such as VoIP.

How to Configure Link Layer Discovery Protocol in Multivendor Networks

Enabling and Disabling LLDP Globally

LLDP is disabled globally by default. This section describes the tasks for enabling and disabling LLDP globally.

Enabling LLDP Globally

Perform this task to enable LLDP globally.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **lldp run**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**lldp run** **Example:** `Router(config)# lldp run`	Enables LLDP globally.
Step 4	**end** **Example:** `Router(config)# end`	Returns the command-line interface (CLI) to privileged EXEC mode.

Disabling LLDP Globally

Perform this task to disable LLDP globally.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no lldp run**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**no lldp run** **Example:** `Router(config)# no lldp run`	Disables LLDP globally.
Step 4	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Disabling and Enabling LLDP on a Supported Interface

LLDP is enabled by default on all supported interfaces. This section describes the tasks for disabling and enabling LLDP on a supported interface.

- Disabling LLDP on a Supported Interface, page 274
- Enabling LLDP on a Supported Interface, page 275

Disabling LLDP on a Supported Interface

Perform this task to disable LLDP on a supported interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no lldp** {**med-tlv-select** *tlv* | **receive** | **transmit**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/1`	Specifies the interface type and number and enters interface configuration mode.
Step 4	**no lldp** {**med-tlv-select** *tlv* | **receive** | **transmit**} **Example:** `Router(config-if)# no lldp receive`	Disables an LLDP-MED TLV or LLDP packet reception on a supported interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Enabling LLDP on a Supported Interface

LLDP information can be transmitted and received only on an interface where LLDP is configured and enabled. Perform this task to enable LLDP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **lldp** {**med-tlv-select** *tlv* | **receive** | **transmit**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/1`	Specifies the interface type and number and enters interface configuration mode.
Step 4	**lldp** {**med-tlv-select** *tlv* | **receive** | **transmit**} **Example:** `Router(config-if)# lldp transmit`	Enables an LLDP-MED TLV or LLDP packet transmission on a supported interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Setting LLDP Packet Hold Time

Hold time is the duration that a receiving device should maintain LLDP neighbor information before aging it. Perform this task to define a hold time for an LLDP-enabled device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **lldp holdtime** *seconds*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**lldp holdtime** *seconds* **Example:** `Router(config)# lldp holdtime 100`	Specifies the hold time.
Step 4	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Setting LLDP Packet Frequency

Perform this task to specify an interval at which the Cisco IOS software sends LLDP updates to neighboring devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **lldp timer** *rate*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**lldp timer** *rate* **Example:** `Router(config)# lldp timer 75`	Specifies the rate at which LLDP packets are sent.
Step 4	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Monitoring and Maintaining LLDP in Multivendor Networks

Perform this task to monitor and maintain LLDP in multivendor networks. This task is optional, and Steps 2 and 3 can be performed in any sequence.

SUMMARY STEPS

1. **enable**
2. **show lldp** [**entry** {***** | *word*} | **errors** | **interface** [**ethernet** *number*]| **neighbors** [**ethernet** *number*| **detail**]| **traffic**]
3. **clear lldp** {**counters** | **table**}
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show lldp** [**entry** {***** | *word*} | **errors** | **interface** [**ethernet** *number*]| **neighbors** [**ethernet** *number*| **detail**]| **traffic**] **Example:** `Router# show lldp entry *`	Displays summarized and detailed LLDP information. **Note** When the **show lldp neighbors** command is issued, if the device ID has more than 20 characters, the ID is truncated to 20 characters in command output because of display constraints.
Step 3	**clear lldp** {**counters** | **table**} **Example:** `Router# clear lldp counters`	Resets LLDP traffic counters and tables to zero.
Step 4	**end** **Example:** `Router# end`	Returns the CLI to user EXEC mode.

Enabling and Disabling LLDP TLVs

LLDP TLV support is enabled by default if LLDP is enabled globally and locally on a supported interface. Specific TLVs, however, can be enabled and suppressed.

- Enabling LLDP TLVs, page 279
- Disabling LLDP TLVs, page 280

Enabling LLDP TLVs

Perform this task to enable an LLDP TLV on a supported interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **lldp tlv-select** *tlv*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/1`	Specifies the interface type and number on which to enable LLDP-MED and enters interface configuration mode.
Step 4	**lldp tlv-select** *tlv* **Example:** `Router(config-if)# lldp tlv-select system-description`	Enables a specific LLDP TLV on a supported interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Disabling LLDP TLVs

Perform this task to disable an LLDP TLV on a supported interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no lldp tlv-select** *tlv*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/1`	Specifies the interface type and number on which to disable LLDP-MED and enters interface configuration mode.
Step 4	**no lldp tlv-select** *tlv* **Example:** `Router(config-if)# no lldp tlv-select system-description`	Disables a specific LLDP TLV on a supported interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Enabling and Disabling LLDP-MED TLVs

LLDP-MED TLV support is enabled by default if LLDP is enabled globally and locally on a supported interface. Specific TLVs, however, can be enabled and suppressed.

- Enabling LLDP-MED TLVs, page 282

- Disabling LLDP-MED TLVs, page 283

Enabling LLDP-MED TLVs

Perform this task to enable a specific LLDP-MED TLV on a supported interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **lldp med-tlv-select** *tlv*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/1`	Specifies the interface type and number on which to enable LLDP-MED and enters interface configuration mode.
Step 4	**lldp med-tlv-select** *tlv* **Example:** `Router(config-if)# lldp med-tlv-select inventory-management`	Enables a specific LLDP-MED TLV on a supported interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Disabling LLDP-MED TLVs

Perform this task to disable a specific LLDP-MED TLV from a supported interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no lldp med-tlv-select** *tlv*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/1`	Specifies the interface type and number on which to disable LLDP-MED and enters interface configuration mode.
Step 4	**no lldp med-tlv-select** *tlv* **Example:** `Router(config-if)# no lldp med-tlv-select inventory-management`	Disables a specific LLDP-MED TLV from a supported interface.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Configuring Location TLV

Perform this task to enable the location TLV on a Cisco device.

Note When you configure location information from various modules, such as CDP, LLDP, and LLDP-MED, you can use the **location prefer** command to configure the priority. For more information on the **location prefer** command, see the *Cisco IOS Network Management Command Reference Guide*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **location** {**admin-tag** *string* | **civic-location identifier** *id* | **elin-location** *string* **identifier** *id*}
4. **exit**
5. **interface** *type number*
6. **location** {**additional-location-information** *word* | **civic-location-id** *id* [**port-location**]| **elin-location-id** *id*}
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**location** {**admin-tag** *string* | **civic-location identifier** *id* | **elin-location** *string* **identifier** *id*} **Example:** `Router(config)# location admin-tag location1`	Specifies the location information for an endpoint and enters civic location configuration mode.

	Command or Action	Purpose
Step 4	**exit** **Example:** `Router(config-civic)# exit`	Returns the CLI to global configuration mode.
Step 5	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0`	Specifies the interface on which you are configuring the location information, and enters interface configuration mode.
Step 6	**location** {**additional-location-information** *word* | **civic-location-id** *id* [**port-location**]| **elin-location-id** *id*} **Example:** `Router(config-if)# location civic-location-id 1 port-location`	Specifies location information for an interface, and enters civic location port configuration mode. • You can configure port-specific information in civic location port configuration mode.
Step 7	**end** **Example:** `Router(config-if-port)# end`	Returns the CLI to privileged EXEC mode.

Configuration Examples for Link Layer Discovery Protocol in Multivendor Networks

Example Configuring LLDP on Two Routers

The following example shows LLDP configurations for two routers in a network. Hold time, a timer value, and TLVs are configured for each router. In each case an assumption is made that the Ethernet interfaces being configured are in the UP state.

```
! Configure LLDP on Router 1 with hold time, timer, and TLV options.
Router1> enable
Router1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)# lldp run
Router1(config)# lldp holdtime 150
Router1(config)# lldp timer 15
Router1(config)# lldp tlv-select port-vlan
Router1(config)# lldp tlv-select mac-phy-cfg
```

```
Router1(config)# interface ethernet 0/0
Router1(config-if)# end
00:08:32: %SYS-5-CONFIG_I: Configured from console by console
! Show the updated running configuration. LLDP is enabled with hold time, timer, and TLV
options configured.
Router1# show running-config
Building configuration...
Current configuration : 1397 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
clock timezone PST -8
ip subnet-zero
!
!
lldp timer 15
lldp holdtime 150
!
!
!
interface Loopback0
 ip address 127.0.0.1 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
!
interface Ethernet0/1
 no ip address
 shutdown
!
interface Ethernet0/2
 no ip address
 shutdown
!
interface Ethernet0/3
 no ip address
 shutdown
!
interface Ethernet1/0
 no ip address
 shutdown
!
interface Ethernet1/1
 no ip address
 shutdown
!
interface Ethernet1/2
 no ip address
 shutdown
!
interface Ethernet1/3
 no ip address
 shutdown
!
interface Serial2/0
 no ip address
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
```

```
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/0
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip classless
!
no ip http server
!
!
!
!
!
control-plane
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
end
! Configure LLDP on Router 2 with hold time, timer, and TLV options.
Router2> enable
Router2# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router2(config)# lldp run
Router2(config)# lldp holdtime 150
Router2(config)# lldp timer 15
Router2(config)# lldp tlv-select port-vlan
Router2(config)# lldp tlv-select mac-phy-cfg
Router2(config)# interface ethernet 0/0
Router2(config-if)# end
00:08:32: %SYS-5-CONFIG_I: Configured from console by console
! Show the updated running configuration on Router 2. LLDP is enabled with hold time,
timer, and TLV options configured.
Router2# show running-config
Building configuration...
Current configuration : 1412 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname R2
```

```
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
clock timezone PST -8
ip subnet-zero
!
!
lldp timer 15
lldp holdtime 150
!
!
!
interface Loopback0
 ip address 127.0.0.1 255.255.255.255
!
interface Ethernet0/0
 no ip address
 shutdown
!
interface Ethernet0/1
 no ip address
 shutdown
!
interface Ethernet0/2
 no ip address
 shutdown
!
interface Ethernet0/3
 no ip address
 shutdown
!
interface Ethernet1/0
 no ip address
 shutdown
!
interface Ethernet1/1
 no ip address
 shutdown
!
interface Ethernet1/2
 no ip address
 shutdown
!
interface Ethernet1/3
 no ip address
 shutdown
!
interface Serial2/0
 no ip address
 no fair-queue
 serial restart-delay 0
!
interface Serial2/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial2/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/0
 no ip address
 shutdown
```

```
 serial restart-delay 0
!
interface Serial3/1
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial3/3
 no ip address
 shutdown
 serial restart-delay 0
!
ip classless
!
no ip http server
!
!
!
!
!
control-plane
!
!
line con 0
 logging synchronous
line aux 0
line vty 0 4
 login
!
end
! After both routers are configured for LLDP, issue the show
 command from each router to view traffic and device information.
Router1# show lldp traffic
LLDP traffic statistics:
    Total frames out: 20
    Total entries aged: 0
    Total frames in: 15
    Total frames received in error: 0
    Total frames discarded: 0
    Total TLVs unrecognized: 0
Router1# show lldp neighbors
Capability codes:
    (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
    (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID           Local Intf     Hold-time  Capability      Port ID
Router2             Et0/0          150        R               Et0/0
Total entries displayed: 1
Router2# show lldp traffic
LLDP traffic statistics:
    Total frames out: 15
    Total entries aged: 0
    Total frames in: 17
    Total frames received in error: 0
    Total frames discarded: 2
    Total TLVs unrecognized: 0
Router2# show lldp neighbors
Capability codes:
    (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
    (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID           Local Intf     Hold-time  Capability      Port ID
Router1             Et0/0          150        R               Et0/0
Total entries displayed: 1
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands: master list of commands with complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS Master Commands List, All Releases
Carrier Ethernet commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Carrier Ethernet Command Reference*
LLDP	Link Layer Discovery Protocol
Comparison of LLDP Media Endpoint Discovery (MED) and Cisco Discovery Protocol	*LLDP-MED and Cisco Discovery Protocol*

Standards

Standard	Title
IEEE 802.1ab	*Station and Media Access Control Connectivity Discovery*

MIBs

MIB	MIBs Link
PTOPO MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2922	Physical Topology MIB

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Link Layer Discovery Protocol in Multivendor Networks

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5 Feature Information for Using Link Layer Discovery Protocol in Multivendor Networks

Feature Name	Releases	Feature Information
IEEE 802.1ab LLDP (Link Layer Discovery Protocol)	12.2(33)SXH	LLDP, standardized by the IEEE as part of 802.1ab, enables standardized discovery of nodes, which in turn facilitates future applications of standard management tools such as SNMP in multivendor networks. The following commands were introduced or modified: **clear lldp**, **lldp** and **show lldp**.
ANSI TIA-1057 LLDP-MED Support	12.2(33)SXH	MED is an LLDP enhancement that was formalized by the TIA for VoIP applications. The Cisco implementation of LLDP is based on the IEEE 802.1ab standard. The following commands were introduced or modified: **lldp**, **lldp**(interface).

Feature Name	Releases	Feature Information
Per Port Location Configuration	12.2(55)SE	The Per Port Location Configuration feature provides a mechanism to configure the location attributes specific to different ports. The following commands were introduced or modified: **location**, **location** (interface), **location prefer**, **show location**, **show nmsp**.
Duplication Location Reporting Issue	12.2(55)SE	The Duplication Location Reporting Issue feature ensures CDP to allow only one wired switch to report the location information by identifying and indicating the location of neighbor ports not to report the duplicate location information of the same device. No commands were introduced or modified by this feature.