Carrier Ethernet Configuration Guide, Cisco IOS Release 12.2SR

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

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

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.

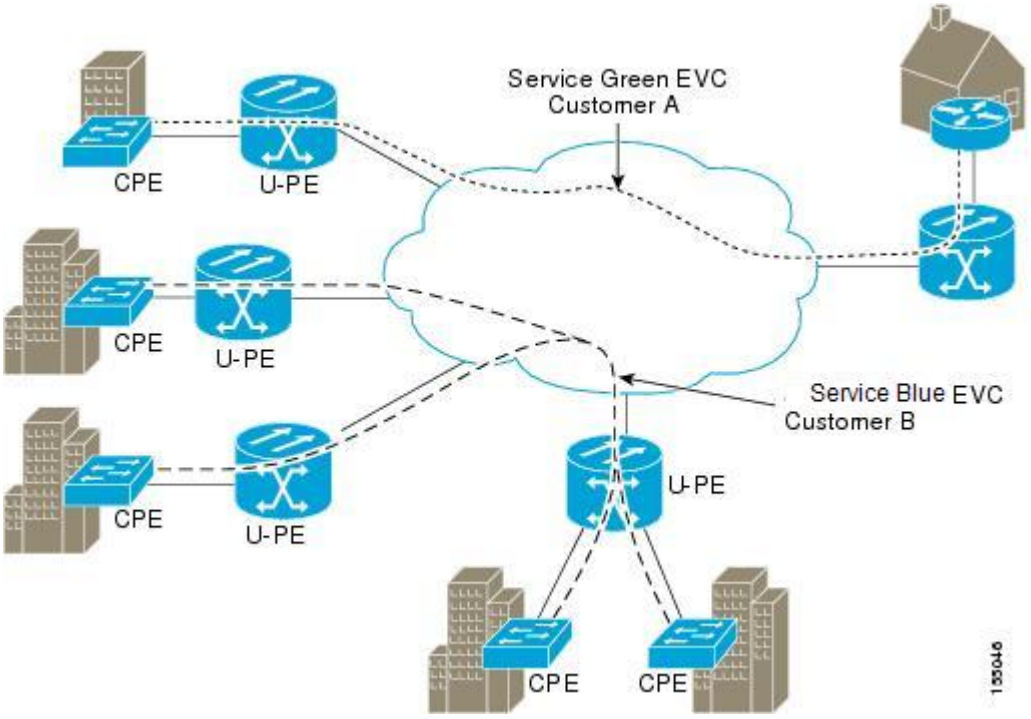
Troubleshooting carrier networks offering Ethernet Layer 2 services is challenging. Customers contract with service providers for end-to-end Ethernet service and service providers may subcontract with operators to provide equipment and networks. Compared to enterprise networks, where Ethernet traditionally 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.



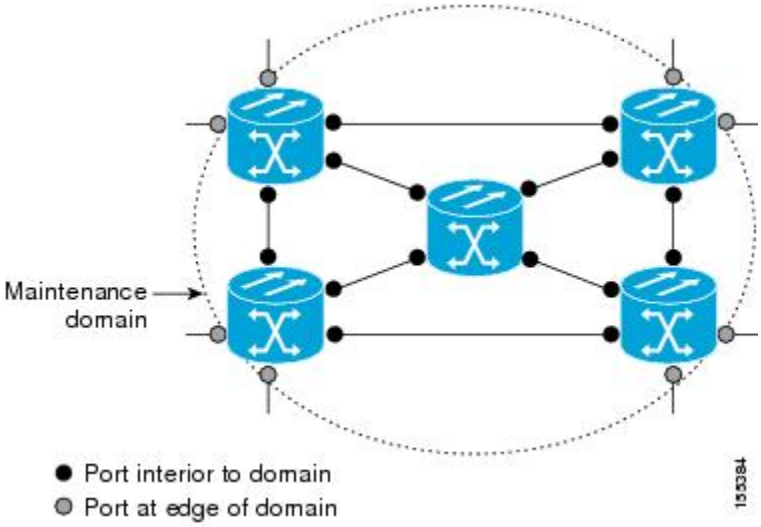
Maintenance Association

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.

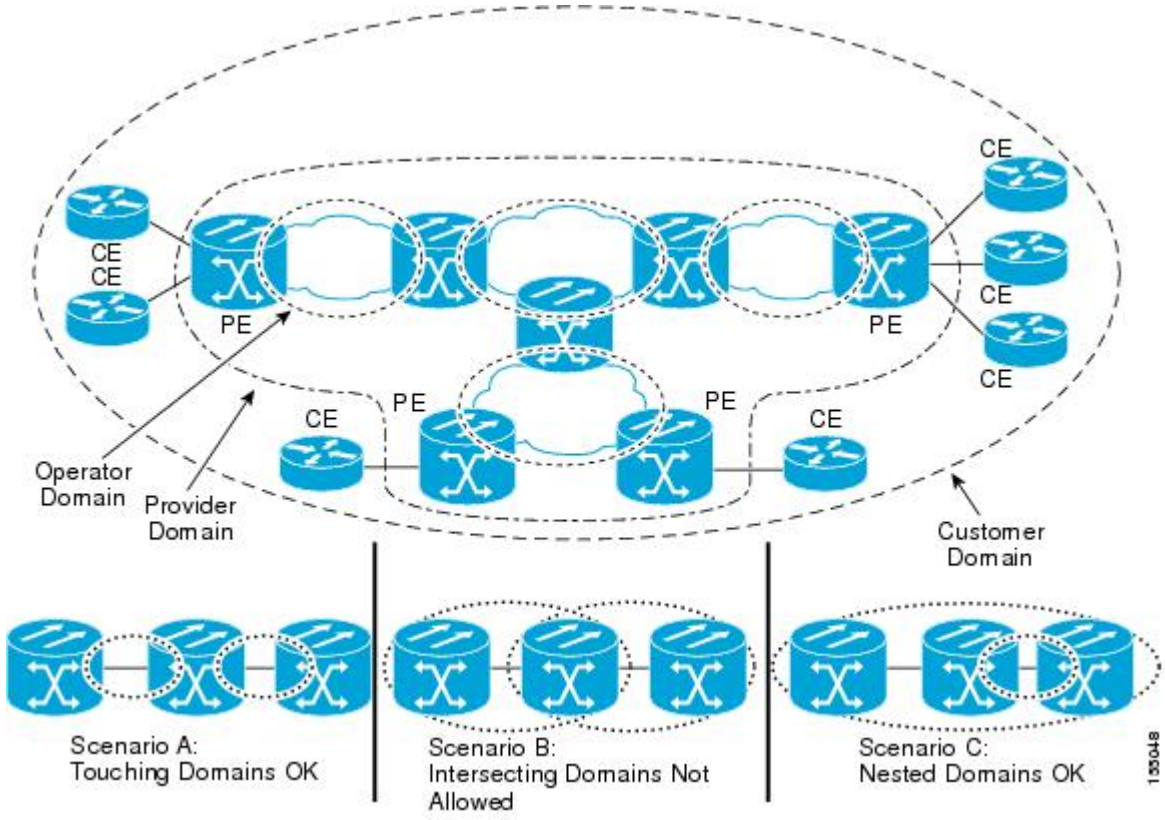
Network designers decide on domains and configurations.

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:

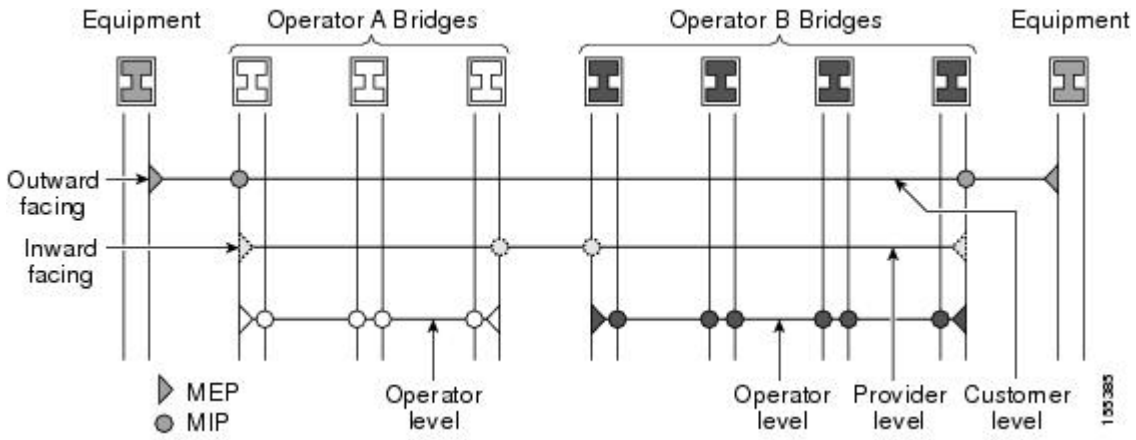
- Can be created manually or dynamically (auto MIPs)
- Dynamically created depending on configured policies at managed objects (MA, maintenance domain, or the default domain level)
- Manual MIPs can be created under an interface and under a service instance within an interface.
- Auto MIP commands can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

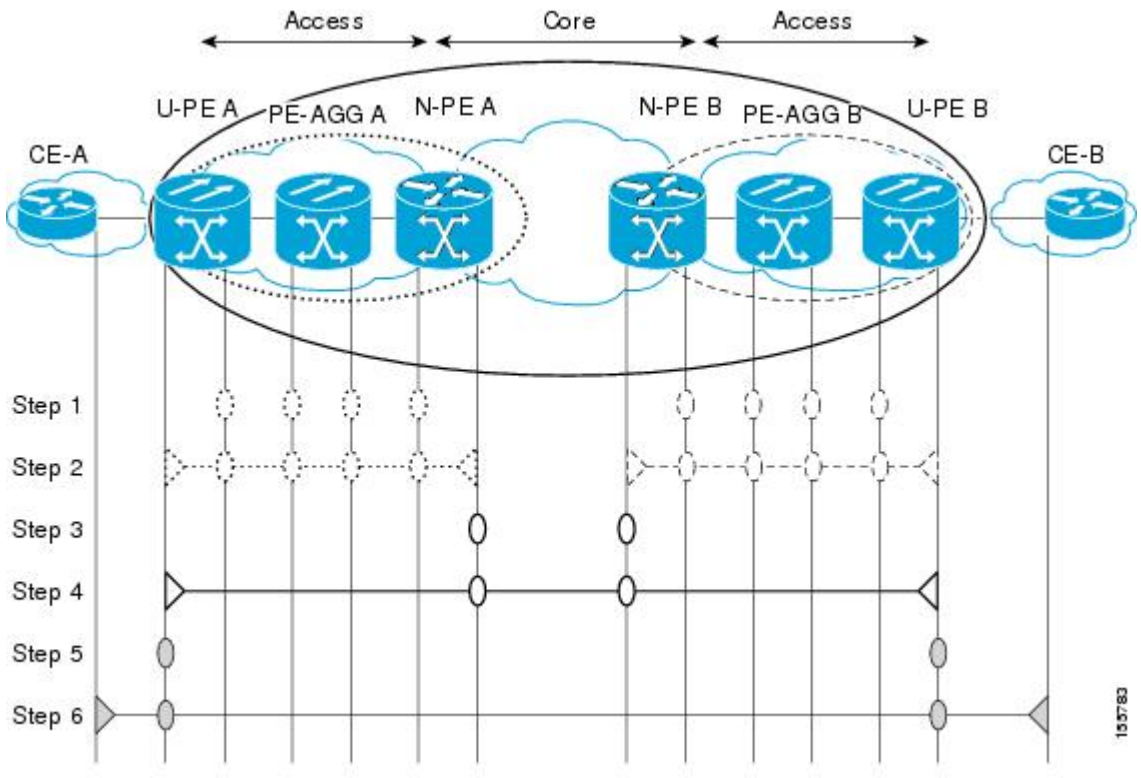
	Command or Action	Purpose
Step 1	Determine operator level MIPs.	Follow these steps: • Starting at lowest operator level domain, assign a MIP at every interface internal to the operator network to be visible to CFM. • Proceed to next higher operator level and assign MIPs. • Verify that every port that 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 179

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 198
- Provisioning the Network on the N-PE B, page 200

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 ITU-T Y.1731 Fault Management Functions in IEEE CFM

This document describes the implementation of the ITU-Y.1731 fault management functions Ethernet Alarm Indication Signal (ETH-AIS) and Ethernet Remote Defect Indication (ETH-RDI) as part of the IEEE Ethernet Connectivity Fault Management (CFM) protocol.

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 ITU-T Y.1731 Fault Management Functions

Business Requirements

- Business and service policies have been established.
- Network topology and network administration have been evaluated.

Technical Requirements

- CFM must be configured and enabled for Y.1731 fault management features to function.
- A server maintenance endpoint (SMEP) is needed to support the ETH-AIS function.
- Maintenance intermediate points (MIPs) must be configured to support AIS messages; they are generated only on an interface on which a MIP is configured.

Restrictions for Configuring ITU-T Y.1731 Fault Management Functions

- Because of a port-ASIC hardware limitation, IEEE CFM cannot coexist with the Per VLAN Spanning Tree (PVST) protocol, and IEEE CFM cannot operate with the following line cards on the same system:
  - FI_WS_X6196_RJ21
  - FI_WS_X6196_RJ45
  - FI_WS_X6548_RJ21
  - FI_WS_X6548_RJ45
- CFM loopback messages are not 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 misconfigure a network and have loopback messages succeed.
  - Security--A malicious device that recognizes devices' MAC addresses and levels may explore a network topology that should be transparent.
- Routed interfaces are supported only in Cisco IOS Release 12.4(11)T.
- IEEE CFM is not fully supported on a Multiprotocol Label Switching (MPLS) provider edge (PE) device. There is no interaction between IEEE 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 restriction:
  - 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 the same way regular data packets are passed. The EoMPLS endpoint interface, however, cannot be a maintenance endpoint (MEP) or an 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.

Information About Configuring ITU-T Y.1731 Fault Management Functions

- ETH-RDI Function, page 154

Continuity Check Messages

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

For more information about CCMs, see the “Continuity Check Messages” section of the "Configuring IEEE Standard-Compliant Ethernet CFM in a Service Provider Network" configuration module.

Server MEPs

Server MEPs (SMEPs) are virtual MEPs that perform two functions--server layer termination for CFM maintenance associations defined at a link or at the transport layer and server-Ethernet adaptation. When a SMEP detects a defect at the server layer, it issues frames containing ETH-AIS information.

Defect Conditions Detected by a MEP

The defect conditions that a MEP detects and subsequently acts upon are the following:

- AIS condition--A MEP receives an AIS frame.
- Dying gasp--An unrecoverable and vendor-specific condition. Dying gasp is generated in the following conditions:
  - Administratively disabling 802.3ah
  - Link down caused by administration down
  - Power failure
  - Reload

Note Administratively disabling 802.3ah does not disrupt traffic and should not generate an AIS. If a Reason field is empty, however, disabling always generates an AIS when Cisco routers and non-Cisco routers are interworking.

A notification about the defect condition may be sent immediately and continuously.

- Loss of continuity (LOC) condition--A MEP stops receiving CCMs from a peer MEP. An LOC condition is a MEP down error.

LOC results when a remote MEP lifetime timer expires and causes an AIS condition for the local MEP. The LOC condition is cleared when connectivity is restored.

- Mismerge condition--A CCM with a correct maintenance level but incorrect maintenance ID indicates that frames from a different service instance are merged with the service instance represented by the receiving MEP’s maintenance ID. A mismerge condition is a cross-connect error.
- RDI condition--A MEP receives a CCM with the RDI field set.
- Signal fail condition--Declared by a MEP or the server layer termination function to notify the SMEP about a defect condition in the server layer. Signal fail conditions are as follows:
  - Configuration error
  - Cross-connect error
  - LOC

- Loop error
- MEP missing
- MEP unknown (same as unexpected MEP)

Signal fail conditions cause AIS defect conditions for the MEP, resulting in the MEP receiving an AIS frame.

A MEP that detects a signal fail condition sends AIS frames to each of the client layer or sublayer maintenance associations.

- Unexpected MEP condition--A CCM with a correct maintenance level, correct maintenance ID, and an unexpected maintenance point ID (MPID) that is the same as the receiving MEP's MPID. An unexpected MEP condition is either a cross-check error or a configuration error.

Determination of an unexpected MPID is possible when a MEP maintains a list of its peer MPIDs. Peer MPIDs must be configured on each MEP during provisioning.

ETH-AIS Function

The ETH-AIS function suppresses alarms when a defect condition is detected at either the server layer or the server sublayer (virtual MEP). Transmission of frames carrying ETH-AIS information can be either enabled or disabled on either a MEP or a SMEP and can be sent at the client maintenance level by either a MEP or SMEP when a defect condition is detected.

SMEPs monitor the entire physical link so that an AIS is generated for each VLAN or server on the network. MEPs monitor VLANs, Ethernet virtual circuits (EVCs), and SMEPs where link up or link down and 802.3ah interworking are supported. A MEP that detects a connectivity fault at a specific level multicasts an AIS in the direction opposite the detected failure at the client maintenance association (MA) level.

An AIS causes a receiving MEP to suppress traps to prevent the network management system (NMS) from receiving an excessive number of redundant traps and also so that clients are asynchronously informed about faults.

In a point-to-point topology, a MEP has a single peer MEP and there is no ambiguity regarding the peer MEP for which it should suppress alarms when it receives ETH-AIS information.

In a multipoint Ethernet topology, a MEP that receives a frame with ETH-AIS information cannot determine which remote peer lost connectivity. The MEP also cannot determine the associated subset of peer MEPs for which it should suppress alarms because the ETH-AIS information does not include that MEP information. Because the MEP cannot determine the affected peer MEPs, it suppresses alarms for all peer MEPs whether or not there is connectivity.

Due to independent restoration capabilities within Spanning Tree Protocol (STP) environments, ETH-AIS is not expected to be applied in these environments; however, ETH-AIS transmission is configurable in STP environments by a network administrator.

- ETH-AIS Transmission Reception and Processing, page 152
- AIS and 802.3ah Interworking, page 153

ETH-AIS Transmission Reception and Processing

Only a MEP or a SMEP can be configured to send frames with ETH-AIS information. When a MEP detects a defect condition, it immediately begins transmitting frames with ETH-AIS information at the configured client maintenance level, which is the level at which the MIP is configured on the interface. Frames are transmitted to peer MEPs in the direction opposite the fault. The first AIS frame must always be

transmitted immediately following the detection of a defect condition, but thereafter frames are transmitted at a frequency based on the configured AIS transmission period. The transmitting MEP continues to transmit frames with ETH-AIS information until the defect condition is removed. The period flag in the frame's header indicates the transmission interval. The default is that a MEP clears a defect condition only if no AIS frames are received within a time period equal to 3.5 times the configured transmission interval.

Note An AIS transmission period of one second is recommended; however, an AIS transmission period of one minute is supported to enable ETH-AIS across all VLANs supported by IEEE CFM.

When a MEP receives a frame with ETH-AIS information, it examines the frame to ensure that the maintenance association level corresponds to its own maintenance association level. The MEP detects the AIS condition and suppresses loss-of-continuity alarms associated with all its peer MEPs. Peer MEPs can resume generating loss-of-continuity alarms only when the receiving MEP exits the AIS condition.

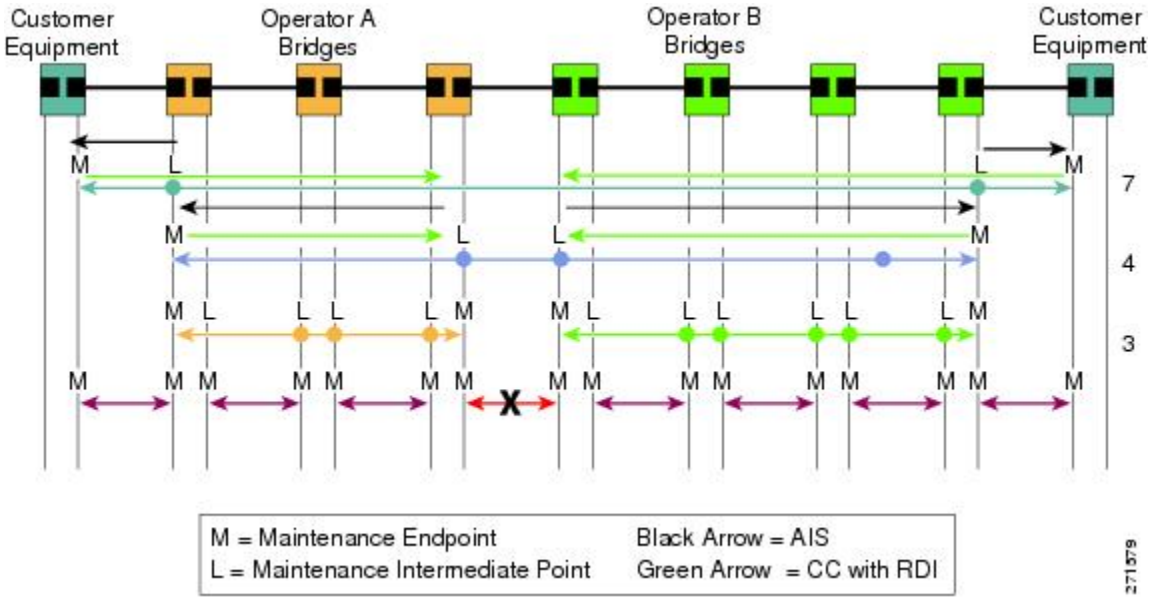
The client layer or client sublayer may consist of multiple maintenance associations that should also be notified to suppress alarms when either a server layer or server sublayer MEP detects a defect condition. The first AIS frame for all client layer or sublayer maintenance associations must be transmitted within one second after the defect condition is detected.

AIS and 802.3ah Interworking

The following conditions impact SMEP AIS conditions:

- By default, link down events cause the SMEP to enter the AIS condition and generate AIS frames for all services at the immediate client maintenance association level.
- Link up events cause the SMEP to exit the AIS state and stop generating AIS frames.
- Local fault detection results from dying gasp, link fault, or critical 802.3ah Remote Fault Indication (RFI). When 802.3ah is reestablished, the SMEP exits the AIS state and stops generating AIS frames.
- Local fault detection due to crossing of a high threshold with a configurable action of error disabling the interface.
- RFI received from a dying gasp, link fault, or critical event.

If a detected fault is due to dying gasp, the link goes down in both directions, creating AIS and RDI frame flow as shown in the figure below.



ETH-RDI Function

The ETH-RDI function is used by a MEP to communicate to its peer MEPs that a defect condition has been encountered. ETH-RDI is used only when ETH-CC transmission is enabled.

ETH-RDI has the following two applications:

- Single-ended fault management--A receiving MEP detects an RDI defect condition, which is correlated with other defect conditions in the MEP and may become the cause of a fault. If ETH-RDI information is not received by a single MEP, there are no defects in the entire MA.
- Contribution to far-end performance monitoring--A defect condition in the far end is used as an input to the performance monitoring process.

A MEP in a defect condition transmits CCMs with ETH-RDI information. A MEP that receives a CCM examines it to ensure that its maintenance association level corresponds to its configured maintenance association level and detects the RDI condition if the RDI field is set. The receiving MEP sets the RDI field in CCMs for the duration of a defect condition, and if the MEP is enabled for CCM transmission, transmits CCMs based on the configured transmission interval. When the defect condition clears, the MEP clears the RDI field in CCMs for subsequent transmissions.

In a point-to-point Ethernet connection, a MEP can clear an RDI condition when it receives the first CCM with the RDI field cleared from its peer MEP. In a multipoint Ethernet connection, a MEP cannot determine the peer MEP with the default condition and can clear an RDI condition only when it receives a CCM with the RDI field cleared from each of its peer MEPs.

The ETH-RDI function is part of continuity checking and is enabled by default. For more information about continuity checking, see the "Configuring IEEE Standard-Compliant Ethernet CFM in a Service Provider Network" configuration module.

How to Configure ITU-T Y.1731 Fault Management Functions

ETH-AIS and ETH-RDI both are enabled by default when CFM is configured, but each can also be manually enabled by a separate command during CFM configuration. Perform these tasks to either disable or enable the functions.

- Disabling the ETH-AIS Function, page 154
- Enabling ETH-AIS for a Single Interface SMEP and Disabling ETH-AIS for All Other Ports, page 156

Disabling the ETH-AIS Function

Perform this task to disable the ETH-AIS function.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm ais link-status global**
4. **disable**
5. **exit**
6. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
7. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
8. **no ais** [**expiry-threshold** | **level** | **period** | **suppress-alarms**]
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	**ethernet cfm ais link-status global** **Example:** `Router(config)# ethernet cfm ais link-status global`	Globally enables AIS generation and enters CFM SMEP AIS configuration mode.
Step 4	**disable** **Example:** `Router(config-ais-link-cfm)# disable`	Disables AIS transmission.
Step 5	**exit** **Example:** `Router(config-ais-link-cfm)# exit`	Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 6	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain PROVIDERDOMAIN level 4`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
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 customer101provider evc customer101provider@101 vlan 101`	Configures a maintenance association within a maintenance domain and enters Ethernet CFM service configuration mode.
Step 8	**no ais** [**expiry-threshold** | **level** | **period** | **suppress-alarms**] **Example:** `Router(config-ecfm-srv)# no ais`	Disables the AIS function for a specific maintenance association.
Step 9	**end** **Example:** `Router(config-ecfm-srv)# end`	Returns the CLI to privileged EXEC mode.

Enabling ETH-AIS for a Single Interface SMEP and Disabling ETH-AIS for All Other Ports

Perform this task to manually enable the ETH-AIS function.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
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. **ais** [**expiry-threshold** *threshold* | **level** *level-id* | **period** *seconds*| **suppress-alarms**]
7. **ais** [**expiry-threshold** *threshold* | **level** *level-id* | **period** *seconds*| **suppress-alarms**]
8. **exit**
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. **ethernet cfm ais link-status global**
12. disable
13. **interface** *type number*
14. **ethernet oam remote-loopback** {**supported** | **timeout** *seconds*}
15. **ethernet cfm mip level** *level-id* [**vlan** {*vlan-id*| *vlan-id - vlan-id*| **,** *vlan-id - vlan-id*}]
16. **ethernet cfm ais link-status** [**level** *level-id*| **period** *seconds*]
17. **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 PROVIDERDOMAIN level 4`	Defines a CFM maintenance domain at a particular maintenance level and enters 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 customer101provider evc customer101provider@101 vlan 101`	Configures a maintenance association within a maintenance domain and enters Ethernet 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	**ais** [**expiry-threshold** *threshold* | **level** *level-id* | **period** *seconds*| **suppress-alarms**] **Example:** `Router(config-ecfm-srv)# ais period 1`	Enables the AIS function for a specific maintenance association.
Step 7	**ais** [**expiry-threshold** *threshold* | **level** *level-id* | **period** *seconds*| **suppress-alarms**] **Example:** `Router(config-ecfm-srv)# ais level 7`	Enables the AIS function for a specific maintenance association.
Step 8	**exit** **Example:** `Router(config-ecfm-srv)# exit`	Returns the CLI to Ethernet CFM configuration mode.
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 customer110provider evc customer110provider@110 vlan 110`	Configures a maintenance association within a maintenance domain and enters Ethernet CFM service configuration mode.

	Command or Action	Purpose
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	**ethernet cfm ais link-status global** **Example:** `Router(config-ecfm-srv)# ethernet cfm ais link-status global`	Globally enables AIS generation and places the CLI in CFM SMEP AIS configuration mode (config-ais-link-cfm) to configure AIS commands for a SMEP.
Step 12	disable **Example:** `Router(config-ais-link-cfm)# disable`	Disables the generation of AIS frames resulting from a link-status change.
Step 13	**interface** *type number* **Example:** `Router(config-ais-link-cfm)# interface ethernet 0/1`	Configures an interface type and enters interface configuration mode.
Step 14	**ethernet oam remote-loopback** {**supported** | **timeout** *seconds*} **Example:** `Router(config-if)# ethernet oam remote-loopback supported`	Enables the support of Ethernet OAM remote loopback operations on an interface or sets a remote loopback timeout period.
Step 15	**ethernet cfm mip level** *level-id* [**vlan** {*vlan-id*| *vlan-id* **-** *vlan-id*| **,** *vlan-id* **-** *vlan-id*}] **Example:** `Router(config-if)# ethernet cfm mip level 4 vlan 101`	Provisions a MIP at a specified maintenance level on an interface.
Step 16	**ethernet cfm ais link-status** [**level** *level-id*| **period** *seconds*] **Example:** `Router(config-if)# ethernet cfm ais link-status`	Enables AIS generation from a SMEP.

	Command or Action	Purpose
Step 17	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Configuration Examples for Configuring ITU-T Y.1731 Fault Management Functions

Example Enabling IEEE CFM on an Interface

The following example shows how to enable IEEE CFM on an interface:

```
!
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
```

Example Enabling AIS

```
The following example shows how to enable AIS:
!
ethernet cfm domain PROVIDER_DOMAIN level 4
 service customer101provider evc customer101provider@101 vlan 101
  continuity-check
  ais period 1
  ais level 7
 service customer110provider evc customer110provider@110 vlan 110
  continuity-check
```

```
!
ethernet cfm ais link-status global
 disable
!
!
interface Ethernet 0/1
 no ip address
 ethernet oam remote-loopback supported
 ethernet oam
 ethernet cfm mip level 4 vlan 1,101,110
 ethernet cfm ais link-status
!
```

Example Show Commands Output

The following sample output from the **show ethernet cfm maintenance-point local detail** command shows the settings for the local MEP:

```
Router# show ethernet cfm maintenance-points local detail

MEP Settings:
-------------
MPID: 2101
DomainName: PROVIDERDOMAIN
Level: 4
Direction: I
Vlan: 101
Interface: Et0/1
CC-Status: Enabled
MAC: aabb.cc03.8410
Defect Condition: AIS
presentRDI: TRUE
AIS-Status: Enabled
AIS Period: 1000(ms)
AIS Expiry Threshold: 3.5
Level to transmit AIS: Default
Suppress Alarm configuration: Enabled
Suppressing Alarms: Yes
```

The following sample output from the **show ethernet cfm smep** command shows the settings for a SMEP:

```
Router# show ethernet cfm smep
SMEP Settings:
--------------
Interface: Ethernet0/0
AIS-Status: Enabled
AIS Period: 60000 (ms)
Level to transmit AIS: 4
Defect Condition: No Defect
```

The following sample output from the **show ethernet cfm smep interface** command shows the settings for a specific interface on a SMEP:

```
Router# show ethernet cfm smep interface ethernet 0/1
SMEP Settings:
--------------
Interface: Ethernet0/1
LCK-Status: Enabled
LCK Period: 60000 (ms)
Level to transmit LCK: Default
AIS-Status: Enabled
AIS Period: 60000 (ms)
Level to transmit AIS: Default
Defect Condition: No Defect
Router#
```

The following sample output from the **show ethernet cfm errors**command shows the Ethernet CFM errors on a device:

```
Router# show ethernet cfm errors
Level    Vlan    MPID    Remote MAC        Reason         Service ID
5        102     -       aabb.cc00.ca10    Receive AIS    service test
```

The following sample output from the **show ethernet cfm maintenance-points remote detail**command shows the detailed information about a specific remote MEP:

```
Router# show ethernet cfm maintenance-points remote detail mpid 66
MAC Address: aabb.cc00.ca10
Domain/Level: PROVIDERDOMAIN/4
EVC: test
MPID: 66 (Can ping/traceroute)
Incoming Port(s): Ethernet0/2
CC Lifetime(sec): 75
Age of Last CC Message(sec): 8
Receive RDI: TRUE
Frame Loss: 0%
CC Packet Statistics: 2/0 (Received/Error)
R1#MAC Address: aabb.cc00.ca10
Domain/Level: PROVIDERDOMAIN/4
EVC: test
MPID: 66 (Can ping/traceroute)
Incoming Port(s): Ethernet0/2
CC Lifetime(sec): 75
Age of Last CC Message(sec): 8
Receive RDI: TRUE
Frame Loss: 0%
CC Packet Statistics: 2/0 (Received/Error)
```

Additional References

Related Documents

Related Topic	Document Title
IEEE CFM	"Configuring IEEE Standard-Compliant Ethernet CFM in a Service Provider Network"
Using OAM	"Using Ethernet Operations, Administration, and Maintenance"
IEEE CFM and Y.1731 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 802.1ag	*802.1ag - Connectivity Fault Management*
IEEE 802.3ah	*Ethernet in the First Mile*
ITU-T	*ITU-T Y.1731 OAM Mechanisms for Ethernet-Based Networks*

MIBs

MIB	MIBs Link
None	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
None	--

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 ITU-T Y.1731 Fault Management Functions

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 ITU-T Y.1731 Fault Management Functions

Feature Name	Releases	Feature Information
Configuring ITU-T Y.1731 Fault Management Functions	15.0(1)XA 12.2(33)SRE 15.1(1)T	The ITU-Y.1731 Fault Management Functions feature adds to IEEE CFM the ETH-AIS and ETH-RDI functions for fault detection, fault verification, and fault isolation in large MANs and WANs. The following commands were introduced or modified: **ais**, **clear ethernet cfm ais**, **disable**(CFM-AIS-link), **ethernet cfm ais link-status**, **ethernet cfm ais link-status global**, **level**(cfm-ais-link), **period**(cfm-ais-link), **show ethernet cfm errors**, **show ethernet cfm maintenance-points local**, **show ethernet cfm maintenance-points remote detail**, **show ethernet cfm smep**.

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.

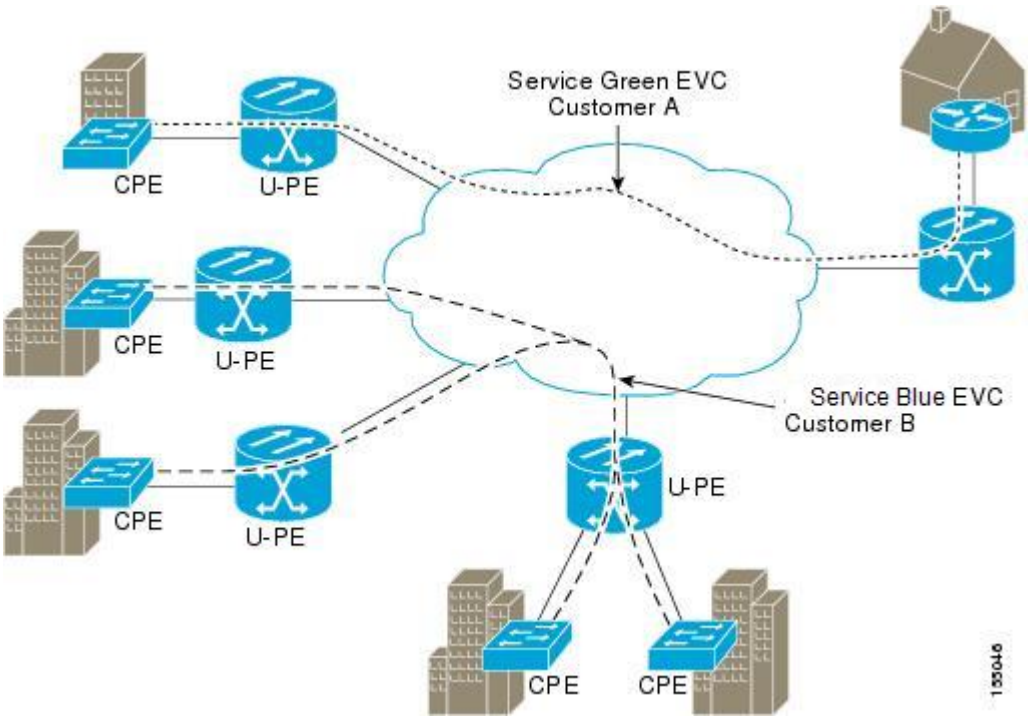
Troubleshooting carrier networks offering Ethernet Layer 2 services is challenging. Customers contract with service providers for end-to-end Ethernet service and service providers may subcontract with operators to provide equipment and networks. Compared to enterprise networks, where Ethernet traditionally 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

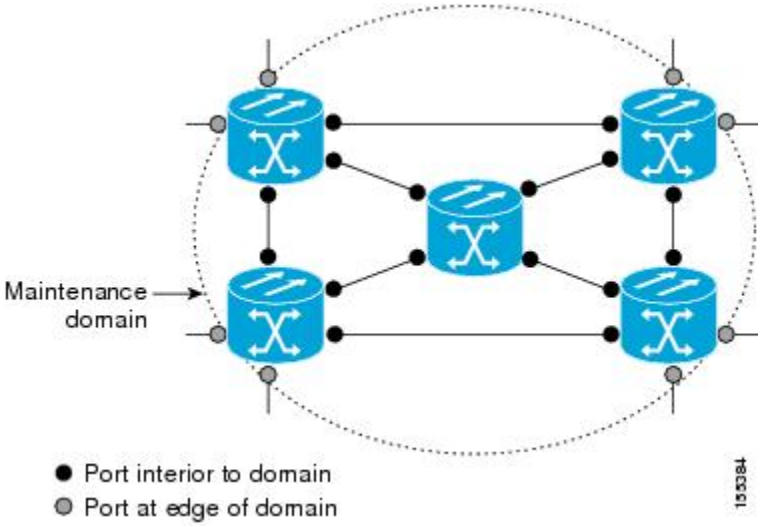
Customer Service Instance

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.

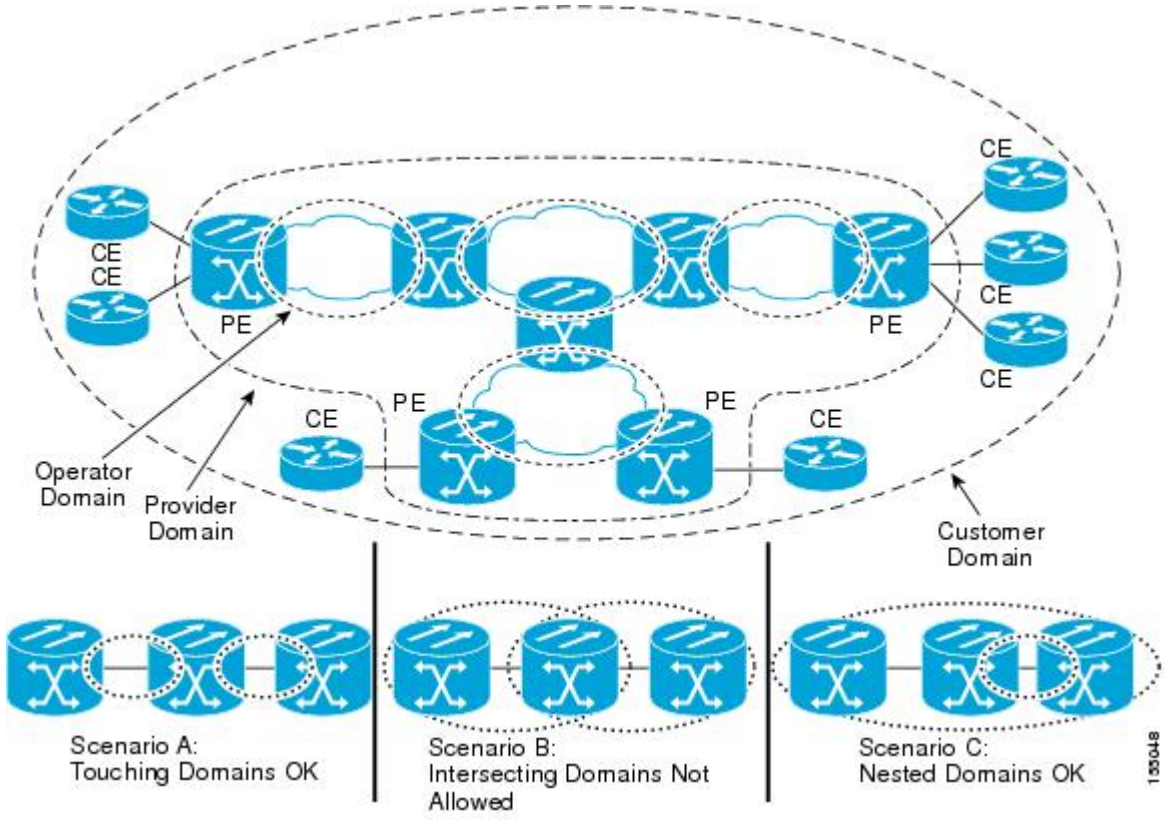


A unique maintenance level in the range of 0 to 7 is assigned to each domain by a network administrator. Levels and domain names are useful for defining the hierarchical relationship that exists among domains. The hierarchical relationship of domains parallels the structure of customer, service provider, 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 170
- Maintenance Intermediate Points, page 171

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:

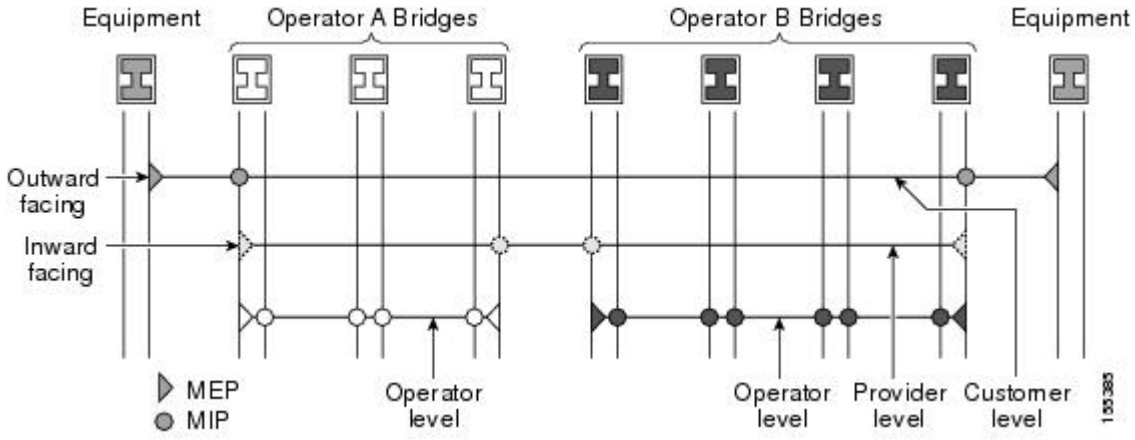
- Per maintenance domain (level) and for all S-VLANs enabled or allowed on a port.
- Internal to a domain, not at the boundary.
- CFM frames received from MEPs and other MIPs are cataloged and forwarded, using both the wire and the relay function.
- All CFM frames at a lower level are stopped and dropped, 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 174
- OAM Manager, page 174
- CFM over Bridge Domains, page 174

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 175

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 179
- Configuring Ethernet OAM Interaction with CFM, page 251

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

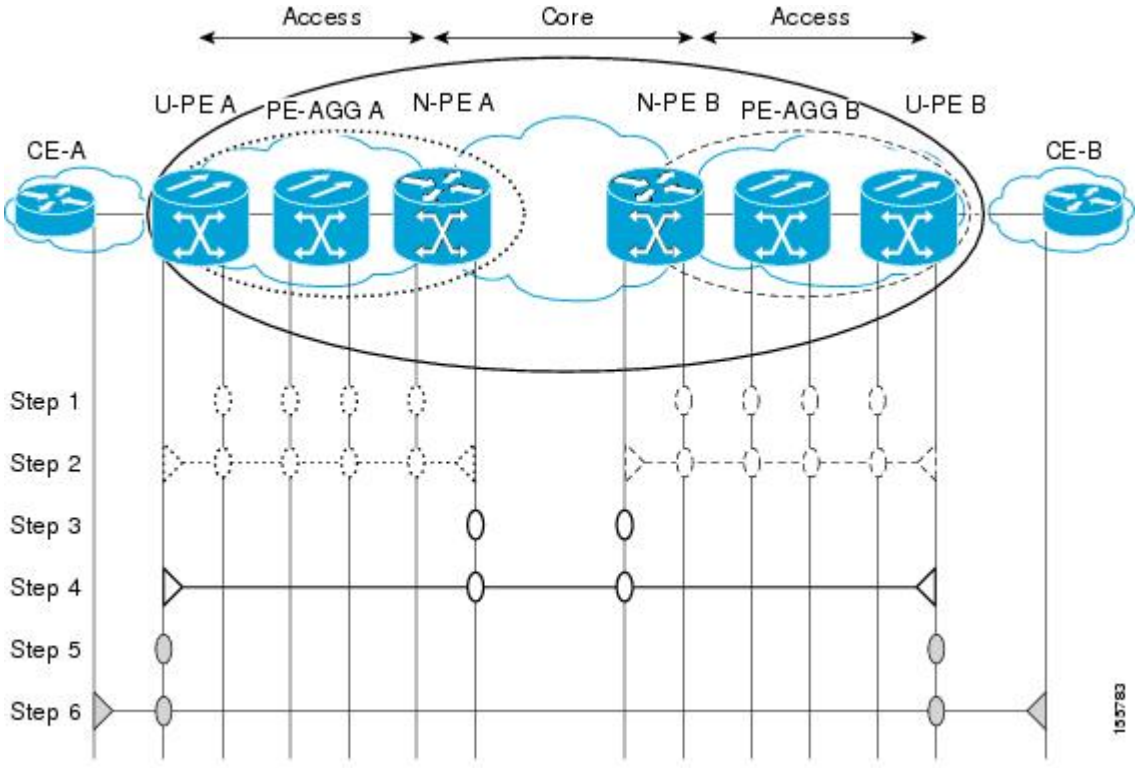
	Command or Action	Purpose
Step 1	Determine operator level MIPs.	Follow these steps: • Starting at lowest operator level domain, assign a MIP at every interface internal to the operator network to be visible to CFM. • Proceed to next higher operator level and assign MIPs. • Verify that every port that 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 179

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 237
- Configuring CFM over Bridge Domains, page 245
- Troubleshooting Tips, page 250

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 for U-PE B, page 69
- Provisioning the Network for CE-B, page 73
- Provisioning the Network on the CE-A, page 180
- Provisioning the Network on the U-PE A, page 182
- Provisioning the Network on the PE-AGG A, page 186
- Provisioning the Network on the N-PE A, page 188
- Provisioning the Network on the CE-B, page 192
- Provisioning the Network on the U-PE B, page 194
- Provisioning the Network on the PE-AGG B, page 198
- Provisioning the Network on the N-PE B, page 200

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 238

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 251
- Enabling Ethernet OAM, page 253

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

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 4 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.

Syslog Support for Ethernet Connectivity Fault Management

The Cisco IOS software system message facility helps to define and report errors and changes in system status. System messages aid customers and Cisco engineers in identifying the types and severities of events and in maintaining and operating Cisco IOS devices. For Ethernet connectivity fault management (CFM), system messages also allow network administrators to develop scripts for effectively configuring and managing the CFM function.

This document describes syslog support for Ethernet CFM and how to enable and disable CFM system messages.

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 Syslog Support for Ethernet Connectivity Fault Management

- Knowledge of the Cisco IOS implementation of Ethernet CFM 802.1ag and of ITU-T Y.1731 fault management functions.

Restrictions for Syslog Support for Ethernet Connectivity Fault Management

- CFM does not support user-configurable actions in response to some events.
- CFM does not support the automatic use of CFM operations such as loopback and linktrace when failures are detected.
- Embedded Event Manager (EEM) does not support Simple Network Management Protocol (SNMP) traps.

Information About Syslog Support for Ethernet Connectivity Fault Management

Syslog Protocol and Messages

Syslog is a delivery method for system messages, typically across an IP network. The term “syslog” is used to describe both the protocol that transfers messages and the messages themselves. Syslog is commonly used for managing computer systems and auditing system security. Syslog is supported by a variety of devices across many platforms. Because of this support, syslog can be used to integrate log data from different types of systems into a central repository.

Syslog messages are text messages less than 1 KB. They can be sent using User Datagram Protocol (UDP), TCP, or both. Messages are not encrypted, but a Secure Sockets Layer (SSL) wrapper can be used to provide a layer of encryption through the SSL or Transport Layer Security (TLS) protocols.

Syslog receivers are called “syslogd,” “syslog daemon,” or “syslog server.”

The syslog protocol and message format are defined in RFC 3164, *The BSD syslog Protocol* .

CFM System Messages

This section describes the types of CFM syslog messages that can be generated and the CFM events that trigger those messages. There are three types of syslog messages:

AIS syslogs

Alarm Indication Signal (AIS) syslog messages can be enabled using the **ethernet cfm logging** command with the **ais** keyword. Following are the AIS syslog messages and corresponding CFM events:

- ENTER_AIS_INT--The interface has entered an AIS defect condition.
- EXIT_AIS_INT--The interface has exited an AIS defect condition.
- ENTER_AIS--An Ethernet CFM maintenance endpoint (MEP) has entered an AIS defect condition.
- EXIT_AIS--An Ethernet CFM MEP has exited an AIS defect condition.

Cisco MIB Alarm syslogs

The same Cisco MIB alarm message definitions apply to both VLAN and Ethernet virtual circuit (EVC) services. Cisco MIB alarm syslog messages can be enabled using the **ethernet cfm logging** command with the **alarm** and **cisco** keywords. Following are the Cisco MIB alarm syslog messages and corresponding CFM events:

- REMOTE_MEP_UP--A continuity check (CC) message is received from an active remove MEP.
- REMOTE_MEP_DOWN--The entry in the CC database corresponding to the MEP times out or the device receives a CC message with a zero hold time.
- CROSS_CONNECTED_SERVICE--The CC message contains a customer service instance (CSI) ID or maintenance association (MA) ID is different from what is configured locally on the device.
- FORWARDING_LOOP--A device is receiving CC messages with its maintenance point ID (MPID) and source MAC address.
- CONFIG_ERROR--A device is receiving a CC message with its MPID but a different source MAC address.
- CROSSCHECK_MEP_MISSING--A configured remote MEP does not come up during the cross-check start timeout interval.
- CROSSCHECK_MEP_UNKNOWN--The remote MEP that is received is not in the configured static list.
- CROSSCHECK_SERVICE_UP--The configured service, either CSI or MA, is up as it receives CC messages from all remote, statically configured MEPs.

IEEE MIB Alarm syslogs

The IEEE MIB alarm syslog message can be enabled using the **ethernet cfm logging** command with the **alarm** and **ieee** keywords. Following is the Cisco MIB alarm syslog message and corresponding CFM event:

- FAULT_ALARM--A fault in the network has occurred.

Syslog Support for Ethernet Connectivity Fault Management

The Syslog Support for Ethernet Connectivity Fault Management (Syslog Support for CFM) feature provides syslog support for CFM notifications that can be used to determine the status of services and of network connectivity. This feature is disabled by default. The command-line interface (CLI) **ethernet cfm logging** command provides the option to either enable or disable all CFM syslogs or to separately enable or disable syslogs for the AIS feature, Cisco MIB alarms, and IEEE MIB alarms.

The Syslog Support for CFM feature must be implemented either on CFM over VLANs or when you use the IEEE 802.1ag on Bridge Domains feature and want to automate diagnostics or implement actions in response to CFM events.

- Benefits of Syslog Support for Ethernet Connectivity Fault Management, page 270

Benefits of Syslog Support for Ethernet Connectivity Fault Management

- Creates a record of events that assists in troubleshooting.
- Establishes a mechanism for leveraging EEM scripts for CFM event notifications.
- Allows control of syslog messages with the CLI **ethernet cfm logging** command.

How to Enable System Message Logging for Ethernet Connectivity Fault Management

Enabling CFM Syslog Messages

CFM syslogs are disabled by default. Perform this task to enable CFM syslog messages.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm logging** [**ais** | **alarm** {**cisco** | **ieee**}]
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	**ethernet cfm logging** [**ais**	**alarm** {**cisco**	**ieee**}] **Example:** `Router(config)# ethernet cfm logging`	Enables all CFM syslog messages.

	Command or Action	Purpose
Step 4	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Disabling CFM Syslog Messages

Perform this task to disable CFM syslog messages.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ethernet cfm logging** [**ais** | **alarm** {**cisco** | **ieee**}]
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	**no ethernet cfm logging** [**ais**	**alarm** {**cisco**	**ieee**}] **Example:** `Router(config)# no ethernet cfm logging`	Disables all CFM syslog messages.
Step 4	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Configuration Examples for System Logging for Ethernet Connectivity Fault Management

Example Enabling All CFM Syslog Messages

The following example shows how to enable all CFM syslog messages:

```
Router> enable
Router# configure terminal
Router(config)# ethernet cfm logging
Router(config)#
```

Example Enabling Cisco MIB Syslog Messages

The following example shows how to enable all Cisco MIB syslog messages:

```
Router> enable
Router# configure terminal
Router(config)# ethernet cfm logging alarm cisco
Router(config)#
```

Example Enabling IEEE MIB Syslog Messages

The following example shows how to enable IEEE MIB syslog messages for VLAN services:

```
Router> enable
Router# configure terminal
Router(config)# ethernet cfm logging alarm ieee
Router(config)#
```

Example Enabling CFM AIS Syslog Messages

The following example shows how to enable syslog messages specific to the CFM AIS feature:

```
Router> enable
Router# configure terminal
Router(config)# ethernet cfm logging ais
Router(config)#
```

Example Disabling All CFM Syslog Messages

The following example shows how to disable all syslog messages:

```
Router> enable
Router# configure terminal
```

```
Router(config)#
no ethernet cfm logging
Router(config)#
```

Additional References

Related Documents

Related Topic	Document Title
Ethernet CFM	Configuring Ethernet Connectivity Fault Management in a Service Provider Network
IEEE 802.3ah	*IEEE 802.3ah Ethernet in the First Mile*
ITU-T Y.1731 fault management functions	Configuring ITU-T Y.1731 Fault Management Functions
Delivering and filtering syslog messages	Reliable Delivery and Filtering for Syslog
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
Cisco IOS Carrier Ethernet commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Carrier Ethernet Command Reference*

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 • CISCO-IEEE-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
RFC 3164	*The BSD syslog Protocol*

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 Syslog Support for Ethernet Connectivity Fault Management

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 Syslog Support for Ethernet Connectivity Fault Management

Feature Name	Releases	Feature Information
Syslog Support for Ethernet Connectivity Fault Management	12.2(33)SRD1	The Syslog Support for Ethernet CFM feature provides syslog support for CFM notifications that can be used to determine the status of services and of network connectivity. This feature must be implemented either when you use the IEEE 802.1ag on Bridge Domains feature or CFM over VLANs or if you are using the IEEE 802.1ag on Bridge Domains feature and want to automate diagnostics or implement actions in response to CFM events. The following commands were introduced or modified: **ethernet cfm logging**.

Configuring ITU-T Y.1731 Fault Management Functions

The ITU-Y.1731 Fault Management Functions feature provides new functions for fault and performance management to serve the needs of service providers in large networks. These new functions extend Ethernet Alarm Indication Signal (ETH-AIS) and Ethernet Remote Defect Indication (ETH-RDI) to include fault detection, fault verification, and fault isolation for large Ethernet Metropolitan-Area Networks (MANs) and Wide-Area Networks (WANs).

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 ITU-T Y.1731 Fault Management Functions

Business Requirements

- Network topology and network administration have been evaluated.
- Business and service policies have been established.

- A Server Maintenance End Point (SMEP) needs to support ETH-AIS function.
- Connectivity Fault Management (CFM) needs to be configured and enabled for Y.1731 Fault Management support.
- Maintenance Intermediate Point (MIP) configuration is required as AIS messages are only generated on the interface which has MIP configured.

Restrictions for Configuring ITU-T Y.1731 Fault Management Functions

- Because of a port-ASIC hardware limitation, CFM cannot coexist with the Per VLAN Spanning Tree (PVST) protocol, and CFM cannot operate with the following line cards on the same system:
  - FI_WS_X6196_RJ45
  - FI_WS_X6196_RJ21
  - FI_WS_X6548_RJ45
  - FI_WS_X6548_RJ21
- CFM loopback messages are not 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 restriction:
  - 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 Maintenance End Point (MEP) or an 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 High Availability features, Non-Stop Forwarding and Stateful Switchover (NSF/SSO) Support, in CFM 802.1ag/1.0d and In Service Software Upgrade (ISSU) Support in CFM 802.1ag/1.0d are not supported on a Customer Edge (CE) device.
- The NSF/SSO Support in CFM 802.1ag/1.0d feature is not supported for the trace route and error databases because the Y.1731 Fault Management HA error database is synchronized between active and standby as both CFM HA and Y.1731 Fault Management were released in Cisco IOS Software Release 12.2SRD.

Information About Configuring ITU-T Y.1731 Fault Management Functions

ETH-AIS General Overview

- When a MEP detects a connectivity fault at a specific level, it will multicast AIS in the direction away from the detected failure at the immediate client Maintenance Association (MA) level.
- AIS is generated by a MEP for each VLAN or server on the network because MEPs monitor the whole physical link. A MEP could be monitoring a VLAN, Ethernet Virtual Connection (EVC), or a SMEP where link up or link down and 802.3ah interworking are supported.
- AIS causes the receiving MEPs to suppress the traps so the Network Management System (NMS) does not receive an excessive number of redundant traps for a particular fault and also so that clients are asynchronously informed regarding faults.
- As AIS cannot determine which remote peer has lost connectivity in a multipoint scenario all peer MEPs enter AIS state and suppress alarms.

Note Use of AIS is not recommended in networks that have independent restoration capability.

ETH-AIS Transmission Reception and Processing Overview

ETH-AIS is used to suppress alarms following detection of defect conditions at the server layer or server sublayer. Transmission of frames with ETH-AIS information can be enabled or disabled on a MEP. Frames with ETH-AIS information can be issued at the client maintenance level by a MEP or SMEP upon detecting defect conditions. For example, the defect conditions may include those in the following sections:

Note Due to independent restoration capabilities provided within Spanning Tree Protocol (STP) environments, ETH-AIS is not expected to be applied in STP environments but ETH-AIS transmission can be configured in STP environments.

For multipoint Ethernet connectivity, a MEP cannot determine the specific server layer or server sublayer that has encountered the defect conditions upon receiving a frame with ETH-AIS information, it also cannot determine the associated subset of its peer MEPs for which it should suppress alarms because the received ETH-AIS information does not contain that information. Therefore, upon reception of a frame with ETH-AIS information, the MEP will suppress alarms for all peer MEPs whether there is still connectivity or not. However, for a point-to-point Ethernet connection, a MEP has only a single peer MEP,

so there is no ambiguity regarding the peer MEP for which it should suppress alarms when it receives the ETH-AIS information.

Only a MEP or an SMEP can be configured to issue frames with ETH-AIS information. Upon detecting a defect condition, the MEP can immediately start transmitting periodic frames with ETH-AIS information at a configured client maintenance level, which in Cisco IOS software is sent at the Maintenance Intermediate Point (MIP) level configured on the interface. A MEP continues to transmit frames with ETH-AIS information until the defect condition is removed. Upon receiving a frame with ETH-AIS information a MEP detects AIS condition and suppresses loss-of-continuity alarms associated with all its peer MEPs. A MEP can only resume loss-of-continuity alarm generation when the MEP exits the AIS condition.

- Signal Fail Conditions When Ethernet Continuity Check Is Enabled, page 280
- AIS Condition When ETH-CC Is Disabled, page 281

Signal Fail Conditions When Ethernet Continuity Check Is Enabled

- Mismerge Condition, page 280
- Unexpected MEP Conditions, page 280

Mismerge Condition

A MEP detects mismerge when it receives a Continuity Check Message (CCM) frame with a correct maintenance level but incorrect maintenance ID that indicates that frames from a different service instance are merged with the service instance represented by the MEP's own maintenance ID.

Note In Cisco IOS Software mismerge condition will be a cross-connect error.

Unexpected MEP Conditions

A MEP detects unexpected MEP conditions when it receives a CCM frame with a correct maintenance level, a correct maintenance ID, and an unexpected MEP ID that includes the MEP's own MEP ID. Determination of unexpected MEP ID is possible when the MEP maintains a list of its peer MEP IDs. A list of peer MEP IDs must be configured on each MEP during provisioning. This defect condition is most likely caused by misconfiguration.

Note In Cisco IOS Software unexpected MEP conditions will be cross-check or configuration errors. A configuration error occurs when the Maintenance Point ID (MPID) received through CCM is the same as the MPID configured on the MEP.

- Unexpected maintenance level condition--A MEP detects unexpected maintenance level when it receives a CCM frame with the incorrect maintenance level.
- Unexpected period condition--A MEP detects unexpected period when it receives a CCM frame with a correct maintenance level, a correct MPID, and a correct MEP ID, but with a period field value different from the MEP's own CCM transmission period.
- Signal fail condition--Signal fail condition may be declared by the server layer termination function to notify the SMEP about a defect condition in the server layer. Signal fail conditions in Cisco IOS Software due to CCM are as follows:

- Cross-connect error
- Configuration error
- Loop error
- MEP unknown
- MEP missing

AIS Condition When ETH-CC Is Disabled

Signal fail conditions will cause AIS defect conditions for the MEP, resulting in the MEPs receiving an AIS frame.

AIS Transmission

Upon detecting a defect condition a MEP will transmit frames to its peer MEPs in the opposite direction to the fault. The frequency of transmission of AIS frames is based on the AIS transmission period. The first AIS frame must always be transmitted immediately following the detection of a defect condition.

Note An AIS transmission period of 1 second is recommended.

The client layer or client sublayer may consist of multiple MAs that should be notified to suppress alarms resulting from defect conditions detected by the server layer or server sublayer MEP. Upon detecting the signal fail condition the MEP will send AIS frames to each of the client layer or sublayer MAs. The first AIS frame for all client layer or sublayer MAs must be transmitted within 1 second of detection of the defect condition.

Note To support ETH-AIS across all 4094 VLANs that CFM supports another AIS transmission period of 1 minute is also supported.

AIS Reception

Upon receiving an AIS frame, a MEP examines it to ensure that the MA level corresponds to its own MA level. The period field indicates the period at which the AIS frames can be expected. Following detection of AIS defect condition, if no AIS frames are received within an interval of 3.5 times the transmission period, the MEP clears the AIS defect condition.

- Dying Gasp Generation, page 281
- AIS Interworking, page 282

Dying Gasp Generation

Dying Gasp is an unrecoverable condition. This type of condition is vendor specific. A notification about the condition may be sent immediately and continuously.

Dying Gasp is generated in following conditions:

- Link down caused by administration down.
- Power failure.
- Reload.

- Administratively disabling 802.3ah.

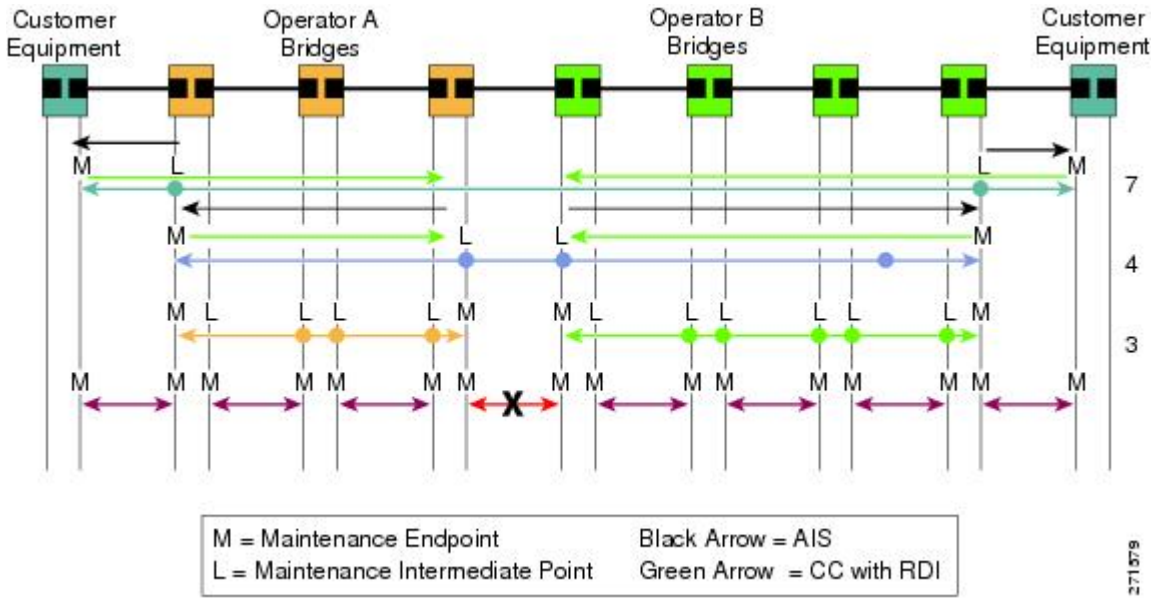
Note Administratively disabling 802.3ah is not traffic disrupting and should not generate AIS. But in the absence of a reason field when interworking with routers other than Cisco routers, disabling will always generate AIS.

AIS Interworking

The following conditions impact SMEP AIS conditions:

- Link down events cause the SMEP to enter AIS condition and generate AIS frames for all services by default at the immediate client MA level.
- Link up events cause the SMEP to exit AIS condition and stop generating AIS frames.
- Local fault detection due to Dying gasp, link fault and critical 802.3ah Remote Fault Indication (RFI). When 802.3ah is reestablished the SMEP exits AIS condition and stops generating AIS frames.
- Local fault detection due to crossing of high threshold whose configurable action is error disabling the interface.
- RFI received from Dying gasp, link fault or critical event.

If the detected fault is due to Dying gasp, the link will go down in both directions creating AIS and RDI frame flow as shown in the figure below.



ETH-RDI

ETH-RDI can be used by a MEP to communicate to its peer MEPs that a defect condition has been encountered. ETH-RDI is used only when Ethernet OAM Continuity Check (ETH-CC) transmission is enabled.

ETH-RDI has the following two applications:

- Single-ended fault management--The receiving MEP detects a RDI defect condition, which gets correlated with other defect conditions in this MEP and may become a fault cause. The absence of received ETH-RDI information in a single MEP indicates the absence of defects in the entire MA.
- Contribution to far-end performance monitoring--It reflects that there was a defect condition in the far end, which is used as an input to the performance monitoring process.

A MEP that is in a defect condition transmits frames with ETH-RDI information. Upon receiving frames with ETH-RDI information, a MEP determines that its peer MEP has encountered a defect condition. However, for multipoint Ethernet connectivity, a MEP, upon receiving frames with ETH-RDI information, cannot determine the associated subset of its peer MEPs with which the MEP transmitting RDI information encounters defect conditions, as the transmitting MEP itself does not always have that information.

CCM Information

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

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 hold time value is 3.5 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.
- Carries the status of the port on which the MEP is configured.

CCM with ETH-RDI Reception

Upon receiving a CCM frame, a MEP examines it to ensure that its MA Level corresponds to its configured MA Level and detects RDI condition if the RDI field is set. For a point-to-point Ethernet connection, a MEP can clear the RDI condition when it receives the first CCM frame from its peer MEP with the RDI field cleared. For multipoint Ethernet connectivity, a MEP can clear the RDI condition when it receives the CCM frames from its entire list of peer MEPs with the RDI fields cleared.

How to Configure ITU-T Y.1731 Fault Management Functions

Y.1731 fault management enhancements consist of ETH-AIS and ETH-RDI. Both enhancements are enabled by default when CFM is configured but each is enabled by a separate command during CFM configuration.

- ETH-AIS is enabled by default by the **ethernet cfm enable**command.
- ETH-RDI is enabled by default by the **ethernet cfm cc enable level command.**

Perform this task to change the default configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm ais link-status global**
4. Do one of the following:
   - **level level-id**
   -
   -
   - disable
5. **period value**
6. **exit**
7. Do one of the following:
   - **ethernet cfm ais domain domain-id [vlan vlan-id | evc evc-name]**
8. **disable**
9. **period value**
10. **level level-id**
11. **expiry-threshold value**
12. **no suppress-alarm**
13. **exit**
14. **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*}
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. **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.

	Command or Action	Purpose
Step 3	**ethernet cfm ais link-status global** **Example:** `Router(config)# ethernet cfm ais link-status global`	Configures AIS specific commands for SMEP and enters config-ais-link-cfm mode.

Step 4	Do one of the following:
	• **level level-id**
	•
	•
	• disable
	
	Example:
	`Router(config-ais-link-cfm)# level 3`
	
	Example:
	
	Example:
	
	Example:
	`Router(config-ais-link-cfm)# disable`
		Configures the maintenance level to send AIS frames transmitted by the SMEP.
		or
		Disables ETH-AIS generation.

Step 5	**period value** **Example:** `Router(config-ais-link-cfm)# period 1`	Configures the SMEP's specific AIS transmission period interval.
Step 6	**exit** **Example:** `Router(config-ais-link-cfm)# exit`	Returns the CLI to global configuration mode.

	Command or Action	Purpose

Step 7	Do one of the following:
	• **ethernet cfm ais domain domain-id [vlan vlan-id | evc evc-name]**
	Example:
	`Router(config)# ethernet cfm ais domain PROVIDER vlan 44`
	Example:
	Example:
	Example:
	`Router(config)# ethernet cfm ais domain XXX evc test`
		Defines a CFM maintenance domain named PROVIDER on VLAN 44 and puts the command in config-ais-mep-cfm submode to configure parameters for all local MEPs belonging to that MA.
		or
		Defines a CFM maintenance domain named XXX on evc test and puts the command in config-ais-mep-cfm submode to configure parameters for all local MEPs belonging to that MA.

Step 8	**disable** **Example:** `Router(config-ais-mep-cfm)# disable`	Disables AIS transmission from 802.03ah.
Step 9	**period value** **Example:** `Router(config-ais-mep-cfm)# period 1`	Configures the SMEP specific AIS transmission period interval.
Step 10	**level level-id** **Example:** `Router(config-ais-mep-cfm)#level 4`	Configures the maintenance level to send AIS frames transmitted by the MEP.
Step 11	**expiry-threshold value** **Example:** `Router(config-ais-mep-cfm)#expiry-threshold 20`	Sets the expiry threshold parameter for the MA.

	Command or Action	Purpose
Step 12	**no suppress-alarm** **Example:** `Router(config-ais-mep-cfm)#no suppress-alarm`	Overrides suppression of redundant alarm.
Step 13	**exit** **Example:** `Router(config-ais-mep-cfm)# exit`	Returns the command to global configuration mode.
Step 14	**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 level 2 on VLAN 100 and enables the sending of traps between MEPs.
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 following parameters for CCMs: • All maintenance levels are to be configured. • All VLANs are to be configured. • The time between CCM transmissions is 20 seconds. • The maximum number of CCMs that can be missed before a MEP is declared down is 3.
Step 16	**exit** **Example:** `Router(config)# exit`	Returns the command to privileged EXEC mode.

How to Configure ITU-T Y.1731 Fault Management Functions

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
   - **ethernet cfm ais link-status**
   -
   -
   - no ethernet cfm ais link-status
5. **ethernet cfm ais link-status period value**
6. **ethernet cfm ais link-status level level-id**
7. **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 GigabitEthernet0/1`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose

Step 4	Do one of the following:
	• **ethernet cfm ais link-status**
	•
	•
	• no ethernet cfm ais link-status
	
	Example:
	`Router(config-if)# ethernet cfm ais link-status`
	
	Example:
	
	Example:
	
	Example:
	`Router(config-if)# no ethernet cfm ais link-status`
		Enables AIS generation from SMEP on interface.
		or
		Disables AIS generation on the interface.

Step 5	**ethernet cfm ais link-status period value** **Example:** `Router(config-if)# ethernet cfm ais link-status period 60`	Sets the AIS transmission period generated by the SMEP on the interface.
Step 6	**ethernet cfm ais link-status level level-id** **Example:** `Router(config-if)# ethernet cfm ais link-status level 4`	Sets the maintenance level to send AIS frames transmitted by the SMEP on the interface.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.

Configuration Examples for Configuring ITU-T Y.1731 Fault Management Functions

Example Enabling Ethernet CFM on an Interface

```
!
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
```

Examples show ethernet cfm Command Output

```
Router# show ethernet cfm maintenance-points local detail

MEP Settings:
-------------
MPID: 2101
DomainName: PROVIDER_DOMAIN
Level: 4
Direction: I
Vlan: 101
Interface: Et0/1
CC-Status: Enabled
MAC: aabb.cc03.8410
Defect Condition: AIS
presentRDI: TRUE
AIS-Status: Enabled
AIS Period: 1000(ms)
AIS Expiry Threshold: 3.5
Level to transmit AIS: Default
Suppress Alarm configuration: Enabled
Suppressing Alarms: Yes
Router# show ethernet cfm smep interface
  Ethernet  IEEE 802.3
Router# show ethernet cfm smep
SMEP Settings:
--------------
Interface: Ethernet0/0
AIS-Status: Enabled
AIS Period: 60000 (ms)
Level to transmit AIS: 4
Defect Condition: No Defect
Router# show ethernet cfm error
Level Vlan      MPID   Remote MAC         Reason            Service ID
5     102       -      aabb.cc00.ca10     Receive AIS       service test
Router# show ethernet cfm maintenance-points remote detail mpid 66
```

```
MAC Address: aabb.cc00.ca10
Domain/Level: PROVIDER_DOMAIN/4
EVC: test
MPID: 66 (Can ping/traceroute)
Incoming Port(s): Ethernet0/2
CC Lifetime(sec): 75
Age of Last CC Message(sec): 8
Receive RDI: TRUE
Frame Loss: 0%
CC Packet Statistics: 2/0 (Received/Error)
R1#MAC Address: aabb.cc00.ca10
Domain/Level: PROVIDER_DOMAIN/4
EVC: test
MPID: 66 (Can ping/traceroute)
Incoming Port(s): Ethernet0/2
CC Lifetime(sec): 75
Age of Last CC Message(sec): 8
Receive RDI: TRUE
Frame Loss: 0%
CC Packet Statistics: 2/0 (Received/Error)
```

Example Syslog AIS Message with Interface Name

```
00:05:39: %ETHER_CFM-6-ENTER_AIS: local mep with mpid 101 level 4 id 7 dir I Interface
Ethernet0/0 enters AIS defect condition
00:05:39: %ETHER_CFM-6-EXIT_AIS: local mep with mpid 101 level 4 id 7 dir I Interface
Ethernet0/0 exitec AIS defect condition
00:05:39: %ETHER_CFM-6-ENTER_AIS_INT: Interface Ethernet0/0 enters AIS defect condition
for outward direction
00:05:39: %ETHER_CFM-6-EXIT_AIS_INT: Interface Ethernet0/0 exited AIS defect condition
for outward direction
```

Additional References

Related Documents

Related Topic	Document Title
Ethernet CFM	*Configuring Ethernet Connectivity Fault Management in a Service Provider Network*
Using OAM	Using Ethernet Operations, Administration, and Maintenance
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

Standards

Standard	Title
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.	--

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 ITU-T Y.1731 Fault Management Functions

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 6 Feature Information for Configuring ITU-T Y.1731 Fault Management Functions

Feature Name	Releases	Feature Information
Configuring ITU-T Y.1731 Fault Management Functions	12.2(33)SRD 12.2(50)SY	The ITU-Y.1731 Fault Management Functions feature provides new functions for fault and performance management to serve the needs of service providers in large networks. These new functions extend Ethernet Alarm Indication Signal (ETH-AIS) and Ethernet Remote Defect Indication (ETH-RDI) to include fault detection, fault verification, and fault isolation for large Ethernet MANs and WANs. The following sections provide information about this feature: The following commands were introduced or modified: **clear ethernet cfm ais , ethernet cfm ais link-status global, show ethernet cfm error, show ethernet cfm maintenance-points remote detail, show ethernet cfm smep interface**.

Layer 2 Access Control Lists on EVCs

The ability to filter packets in a modular and scalable way is important for both network security and network management. Access Control Lists (ACLs) provide the capability to filter packets at a fine granularity. In Metro Ethernet networks, ACLs are directly applied on Ethernet virtual circuits (EVCs).

Layer 2 Access Control Lists on EVCs is a security feature that allows packet filtering based on MAC addresses. This module describes how to implement ACLs on EVCs.

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 Layer 2 Access Control Lists on EVCs

- Knowledge of how service instances must be configured.
- Knowledge of extended MAC ACLs and how they must be configured.

Restrictions for Layer 2 Access Control Lists on EVCs

- A maximum of 16 access control entries (ACEs) are allowed for a given ACL.

- Only 256 different or unique Layer 2 ACLs can be configured on a line card. (More than 256 ACLs can be configured on a router.)
- Layer 2 ACLs function inbound only.
- Current Layer 2 ACLs provide Layer 3 filtering options in permit and deny rules. Options that are not relevant to service instances are ignored.

Information About Layer 2 Access Control Lists on EVCs

EVC

An EVC as defined by the Metro Ethernet Forum is a port-level point-to-point or multipoint-to-multipoint Layer 2 circuit. It is an end-to-end representation of a single instance of a Layer 2 service being offered by a provider to a customer. It embodies the different parameters on which the service is being offered. A service instance is the instantiation of an EVC on a given port on a given router.

Ethernet virtual connection services (EVCS) uses EVCs and service instances to provide Layer 2 switched Ethernet services. The EVC status can be used by a customer edge (CE) device either to find an alternative path in to the service provider network or, in some cases, to revert to a backup path over Ethernet or over another alternative service such as Frame Relay or ATM.

For information about the Metro Ethernet Forum standards, see the Standards table in the Additional References section.

Relationship Between ACLs and Ethernet Infrastructure

The following points capture the relationship between ACLs and Ethernet Infrastructure (EI):

- ACLs can be directly applied on an EVC using the command-line interface (CLI). An ACL is applied to a service instance, which is the instantiation of an EVC on a given port.
- One ACL can be applied to more than one service instance at any time.
- One service instance can have one ACL at most applied to it at any time. If a Layer 2 ACL is applied to a service instance that already has a Layer 2 ACL, the new one replaces the old one.
- Only named ACLs can be applied to service instances. The command syntax ACLs is retained; the **mac access-list extended** command is used to create an ACL.
- The **show ethernet service instance** command can be used to provide details about ACLs on service instances.

How to Configure Layer 2 Access Control Lists on EVCs

Creating a Layer 2 ACL

Perform this task to create a Layer 2 ACL with a single ACE.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mac access-list extended** *name*
4. **permit** {{*src-mac mask* | **any**} {*dest-mac mask* | **any**} [*protocol* [**vlan** *vlan*] [*cos value*]]}

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	**mac access-list extended** *name* **Example:** `Router(config)# mac access-list extended test-12-acl`	Defines an extended MAC ACL and enters mac access list control configuration mode.
Step 4	**permit** {{*src-mac mask* | **any**} {*dest-mac mask* | **any**} [*protocol* [**vlan** *vlan*] [*cos value*]]} **Example:** `Router(config-ext-macl)# permit 00aa.00bb.00cc 0.0.0 any`	Allows forwarding of Layer 2 traffic if the conditions are matched. Creates an ACE for the ACL.

Applying a Layer 2 ACL to a Service Instance

Perform this task to apply a Layer 2 ACL to a service instance. Note that packet filtering takes place only after the ACL has been created and applied to the service instance.

Before applying an ACL to a service instance, you must create it using the **mac access-list extended command. See the “Creating a Layer 2 ACL” section on page 3 .**

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance** *id* ethernet
5. **encapsulation dot1q vlan-id**
6. **mac access-group** *access-list-name* in

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 1/0/0`	Specifies the type and location of the interface to configure, where: • *type* --Specifies the type of the interface. • *number* --Specifies the location of the interface.
Step 4	**service instance** *id* ethernet **Example:** `Router(config-if)# service instance 100 ethernet`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 5	**encapsulation dot1q vlan-id** **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.

	Command or Action	Purpose
Step 6	**mac access-group** *access-list-name* in **Example:** `Router(config-if-srv)# mac access-group test-12-acl in`	Applies a MAC ACL to control incoming traffic on the interface.

Configuring a Layer 2 ACL with ACEs on a Service Instance

Perform this task to configure the same ACL with three ACEs and stop all other traffic on a service instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mac access-list extended** *name*
4. **permit** {*src-mac mask* | **any**} {*dest-mac mask* | **any**}
5. **permit** {*src-mac mask* | **any**} {*dest-mac mask* | **any**}
6. **permit** {*src-mac mask* | **any**} {*dest-mac mask*} | **any**}
7. **deny any any**
8. **exit**
9. **interface** *type number*
10. **service instance** *id* **ethernet**
11. **encapsulation dot1q vlan-id**
12. **mac access-group** *access-list-name* **in**

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	**mac access-list extended** *name* **Example:** `Router(config)# mac access list extended test-12-acl`	Defines an extended MAC ACL and enters mac access control list configuration mode.
Step 4	**permit** {*src-mac mask* | **any**} {*dest-mac mask* | **any**} **Example:** `Router(config-ext-macl)# permit 00aa.bbcc.ddea 0.0.0 any`	Allows forwarding of Layer 2 traffic if the conditions are matched. This creates an ACE for the ACL.
Step 5	**permit** {*src-mac mask* | **any**} {*dest-mac mask* | **any**} **Example:** `Router(config-ext-macl)# permit 00aa.bbcc.ddeb 0.0.0 any`	Allows forwarding of Layer 2 traffic if the conditions are matched. This creates an ACE for the ACL.
Step 6	**permit** {*src-mac mask* | **any**} {*dest-mac mask*} | **any**} **Example:** `Router(config-ext-macl)# permit 00aa.bbcc.ddec 0.0.0 any`	Allows forwarding of Layer 2 traffic if the conditions are matched. This creates an ACE for the ACL.
Step 7	**deny any any** **Example:** `Router(config-ext-macl)# deny any any`	Prevents forwarding of Layer 2 traffic except for the allowed ACEs.
Step 8	**exit** **Example:** `Router(config-ext-macl)# exit`	Exits the current command mode and returns the CLI to global configuration mode.
Step 9	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Specifies the interface.

	Command or Action	Purpose
Step 10	**service instance** *id* **ethernet** **Example:** `Router(config-if)# service instance 200 ethernet`	Configures an Ethernet service instance on an interface and enters service instance configuration mode.
Step 11	**encapsulation dot1q vlan-id** **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to be used to map ingress dot1q frames on an interface to the appropriate service instance.
Step 12	**mac access-group** *access-list-name* **in** **Example:** `Router(config-if-srv)# mac access-group test-12-acl in`	Applies a MAC ACL to control incoming traffic on the interface.

Verifying the Presence of a Layer 2 ACL on a Service Instance

Perform this task to verify that a Layer 2 ACL is present on an EVC. This verification task can be used after an ACL has been configured to confirm its presence.

SUMMARY STEPS

1. **enable**
2. configure terminal
3. **show ethernet service instance id** *id* **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	configure terminal **Example:** `Router# show ethernet service instance id 100 interface gigabitethernet 3/0/1 detail`	Enters global configuration mode.

Command or Action	Purpose
Step 3 **show ethernet service instance id** *id* **interface** *type number* detail **Example:** `Router# show ethernet service instance id 100 interface gigabitethernet 3/0/1 detail`	Displays detailed information about Ethernet customer service instances.

Configuration Examples for Layer 2 Access Control Lists on EVCs

Example Applying a Layer 2 ACL to a Service Instance

The following example shows how to apply a Layer 2 ACL called mac-20-acl to a service instance. The ACL has five permitted ACEs and all other traffic is not allowed.

```
enable
configure terminal
 mac access-list extended mac-20-acl

 permit 00aa.bbcc.adec 0.0.0 any

 permit 00aa.bbcc.bdec 0.0.0 any

 permit 00aa.bbcc.cdec 0.0.0 any

 permit 00aa.bbcc.edec 0.0.0 any

 permit 00aa.bbcc.fdec 0.0.0 any

 deny any any
 exit
interface gigabitethernet 10/0/0
 service instance 100 ethernet
 encapsulation dot1q 100
 mac access-group mac-20-acl in
```

Example Applying a Layer 2 ACL to Three Service Instances on the Same Interface

The following example shows how to apply a Layer 2 ACL called mac-07-acl to three service instances on the same interface:

```
enable
configure terminal
mac access-list extended mac-07-acl

permit 00aa.bbcc.adec 0.0.0 any

permit 00aa.bbcc.bdec 0.0.0 any

permit 00aa.bbcc.cdec 0.0.0 any

deny any any
exit
interface gigabitethernet 10/0/0
service instance 100 ethernet
encapsulation dot1q 100
mac access-group mac-07-acl in
service instance 101 ethernet
encapsulation dot1q 101
mac access-group mac-07-acl in
service instance 102 ethernet
encapsulation dot1q 102
mac access-group mac-07-acl in
```

Example Creating a Layer 2 ACL with ACEs

The following example shows how to create a Layer 2 ACL called mac-11-acl with two permitted ACEs:

```
enable
configure terminal
mac access-list extended mac-11-acl
permit 00aa.00bb.00cc 1a11.0101.11c1 any
permit 00aa.00bb.00cc 1a11.0101.11c2 any
```

Example Displaying the Details of a Layer 2 ACL on a Service Instance

The following sample output displays the details of a Layer 2 ACL called test-acl on a service instance.

```
Router# show ethernet service instance id 100 interface ethernet0/0 detail
Service Instance ID: 100
L2 ACL (inbound): test-acl
Associated Interface: Ethernet0/0
Associated EVC: test
L2protocol drop
CEVlans:
Interface Dot1q Tunnel Ethertype: 0x8100
State: Up
L2 ACL permit count: 10255
L2 ACL deny count: 53
```

The table below describes the significant fields in the output.

Table 7 show ethernet service instance Field Descriptions

Field	Description
Service Instance ID	Displays the service instance ID.
L2 ACL (inbound):	Displays the ACL name.
Associated Interface:	Displays the interface details of the service instance.
Associated EVC:	Displays the EVC with which the service instance is associated.
CEVlans:	Displays details of the associated VLAN ID.
State:	Displays whether the service instance is in an up or down state.
L2 ACL permit count:	Displays the number of packet frames allowed to pass on the service instance by the ACL.
L2 ACL deny count	Displays the number of packet frames not permitted to pass on the service instance by the ACL.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS 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
MEF 6.1	Metro Ethernet Services Definitions Phase 2 (PDF 6/08)
MEF 10.1	Ethernet Services Attributes Phase 2 (PDF 10/06)

MIBs

MIB	MIBs Link
• None	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 release.	--

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 Layer 2 Access Control Lists on EVCs

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 8 Feature Information for Layer 2 Access Control Lists on EVCs

Feature Name	Releases	Feature Information
Layer 2 Access Control Lists on EVCs	12.2(33)SRD 15.0(1)S	The Layer 2 Access Control Lists on EVCs feature introduces ACLs on EVCs. • The following commands were introduced or modified: **interface, mac access-group in**, **mac access-list extended, show ethernet service instance**.

IEEE 802.1s on Bridge Domains

The IEEE 802.1s on Bridge Domains feature enables Multiple Spanning Tree (MST) on Ethernet Virtual Circuits (EVCs).

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 IEEE 802.1s on Bridge Domains

- MST must be configured.

Restrictions for IEEE 802.1s on Bridge Domains

- Service instances on a port-channel are not supported on Cisco 7600 series routers.
- Service instances with “encapsulation default” are not supported.
- Service instances with “encapsulation untagged” without the dot1q option are not supported.
- Service instances with “encapsulation priority-tagged” are not supported.

Information About IEEE 802.1s on Bridge Domains

EVC

An EVC as defined by the Metro Ethernet Forum is a port-level point-to-point or multipoint-to-multipoint Layer 2 circuit. It is an end-to-end representation of a single instance of a Layer 2 service being offered by a provider to a customer. An EVC embodies the different parameters on which the service is being offered. A service instance is the instantiation of an EVC on a specified port.

Service instances are configured under a port channel. The traffic, carried by the service instance is load balanced across member links. Service instances under a port channel are grouped and each group is associated with one member link. Ingress traffic for a single EVC can arrive on any member of the bundle. All egress traffic for a service instance uses only one of the member links. Load balancing is achieved by grouping service instances and assigning them to a member link.

Ethernet virtual connection services (EVCS) uses the concepts of EVCs and service instances to provide Layer 2 switched Ethernet services. EVC status can be used by a Customer Edge (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.

For information about the Metro Ethernet Forum standards, see the Standards table in the Additional References section.

MST and STP

Spanning Tree Protocol (STP) is a Layer 2 link-management protocol that provides path redundancy while preventing undesirable loops in the network. For a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. STP operation is transparent to end stations, which cannot detect whether they are connected to a single VLAN segment or to a switched LAN of multiple segments.

Cisco 7600 series routers use STP (the IEEE 802.1D bridge protocol) on all VLANs. By default, a single instance of STP runs on each configured VLAN (provided you do not manually disable STP). You can enable and disable STP on a per-VLAN basis.

MST maps multiple VLANs into a spanning tree instance, with each instance having a spanning tree topology independent of other spanning tree instances. This architecture provides multiple forwarding paths for data traffic, enables load balancing, and reduces the number of spanning tree instances required to support many VLANs. MST improves the fault tolerance of the network because a failure in one instance (a forwarding path) does not affect other instances.

To participate in MST instances, routers must be consistently configured with the same MST configurations. A collection of interconnected routers that have the same MST configuration forms an MST region. For two or more routers to be in the same MST region, they must have the same VLAN-to-instance mapping, the same configuration revision number, and the same MST name.

The MST configuration controls the MST region to which each router belongs. The configuration includes the name of the region, the revision number, and the MST VLAN-to-instance assignment map.

A region can have one or multiple members with the same MST configuration; each member must be capable of processing Rapid Spanning Tree Protocol (RSTP) bridge protocol data units (BPDUs). There is

no limit to the number of MST regions in a network, but each region can support up to 65 spanning tree instances. Instances can be identified by any number in the range from 0 to 4094. You can assign a VLAN to only one spanning tree instance at a time.

MST on Service Instances with Bridge Domains

The IEEE 802.1s on Bridge Domains feature uses VLAN IDs for service-instance-to-MST-instance mapping. EVC service instances with the same VLAN ID (the outer VLAN IDs in the QinQ case) as the one in a particular MST instance will be mapped to that MST instance.

EVC service instances can have encapsulations with a single tag as well as double tags. In the case of double tag encapsulations, the outer VLAN ID is used for the MST instance mapping, and the inner VLAN ID is ignored.

Because MST requires bridge ports, you must configure a bridge domain for service instances to participate in the MST instances. Additionally, because MST runs by sending untagged BPDUs on the wire, independently of any VLAN, a native VLAN is required on the interface with EVC service instances. By default, switch ports have a native VLAN. However, if the port is not a switch port, you must specify a native VLAN using an EVC service instance.

Because a VLAN ID is required for EVC service-instance-to-MST-instance mapping, the following EVC service instances without any VLAN IDs in the encapsulation are not supported:

- Untagged (encapsulation untagged)
- Priority-tagged (encapsulation priority-tagged)
- Default (encapsulation default)

How to Configure IEEE 802.1s on Bridge Domains

Configuring MST on EVC Bridge Domains

Perform this task to configure MST on EVC bridge domains:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *subslot* / *port* [**.** *subinterface-number*] or**interface tengigabitethernet** *slot* / *subslot* / *port*[**.** *subinterface-number*]
4. **service instance** *id* **ethernet** [*evc-id*]
5. **encapsulation dot1q** *vlan-id* [**native**]
6. **bridge-domain** *bridge-id* [**split-horizon** [**group** *group-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	**interface gigabitethernet** *slot* / *subslot* / *port* [**.** *subinterface-number*] or**interface tengigabitethernet** *slot* / *subslot* / *port*[**.** *subinterface-number*]
	
	Example:
	`Router(config)# interface gigabitethernet 4/0/0`
	`or`
	`Router(config)# interface tengigabitethernet 4/0/0`
		Specifies the Gigabit Ethernet or the Ten Gigabit Ethernet interface to configure,

Step 4	**service instance** *id* **ethernet** [*evc-id*] **Example:** `Router(config-if)# service instance 101 ethernet`	Creates a service instance (an instance of an EVC) on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* [**native**] **Example:** `Router(config-if-srv)# encapsulation dot1q 13`	Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.
Step 6	**bridge-domain** *bridge-id* [**split-horizon** [**group** *group-id*]] **Example:** `Router(config-if-srv)# bridge-domain 12`	Binds the service instance to a bridge domain instance.

• Troubleshooting Tips, page 310

Troubleshooting Tips

The following commands can be used to troubleshoot MST configurations on EVC bridge domains.

- **debug ethernet l2ctrl**
- **debug l2ctrl**

Configuration Examples for IEEE 802.1s on Bridge Domains

Example Configuring MST on EVC Bridge Domains

In the following example, the two interfaces participate in MST instance 0, the default instance to which all VLANs are mapped:

```
Router# enable
Router# configure terminal
Router(config)# interface gigabitethernet 4/0/0
Router(config-if)# service instance 1 ethernet
Router(config-if-srv)# encapsulation dot1q 2
Router(config-if-srv)# bridge-domain 100
Router(config-if-srv)# interface gigabitethernet 4/0/3
Router(config-if)# service instance 1 ethernet
Router(config-if-srv)# encapsulation dot1q 2
Router(config-if-srv)# bridge-domain 100
Router(config-if-srv)# end
```

Issue the following command to verify:

```
Router# show spanning-tree vlan 2
MST0
    Spanning tree enabled protocol mstp
    Root ID         Priority         32768
                 Address         0009.e91a.bc40
                 This bridge is the root
                 Hello Time     2 sec     Max Age 20 sec     Forward Delay 15 sec
    Bridge ID         Priority         32768 (priority 32768 sys-id-ext 0)
                  Address         0009.e91a.bc40
                 Hello Time     2 sec     Max Age 20 sec     Forward Delay 15 sec
Interface                          Role Sts Cost                          Prio.Nbr     Type
------------------                 ---- --- ---------
--------     --------------------------------
Gi4/0/0                         Desg FWD 20000                         128.1537     P2p
Gi4/0/3                         Back BLK 20000                         128.1540     P2p
```

In the following example, interface gigabitethernet 4/0/0 and interface gigabitethernet 4/0/3 are connected back to back. Each has a service instance attached to it. The service instance on both interfaces has an encapsulation VLAN ID of 2. Changing the VLAN ID from 2 to 8 in the encapsulation directive for the service instance on interface gi4/0/0 stops the MSTP from running in the MST instance to which the old VLAN is mapped and starts the MSTP in the MST instance to which the new VLAN is mapped:

```
Router(config-if)# interface gigabitethernet 4/0/0
Router(config-if)# service instance 1 ethernet
Router(config-if-srv)# encapsulation dot1q 8
Router(config-if-srv)# end
```

Issue the following command to verify:

```
Router# show spanning-tree vlan 2
MST1
    Spanning tree enabled protocol mstp
    Root ID             Priority             32769
                Address             0009.e91a.bc40
                This bridge is the root
```

```
                Hello Time     2 sec     Max Age 20 sec     Forward Delay 15 sec
    Bridge ID             Priority            32769 (priority 32768 sys-id-ext 1)
                Address             0009.e91a.bc40
                Hello Time     2 sec     Max Age 20 sec     Forward Delay 15 sec
Interface                     Role Sts Cost                          Prio.Nbr     Type
------------------     ---- --- ---------
--------     --------------------------------
Gi4/0/3                    Desg FWD 20000                         128.1540     P2p
Router# show spanning-tree vlan 8
MST2
    Spanning tree enabled protocol mstp
    Root ID             Priority             32770
                Address             0009.e91a.bc40
                This bridge is the root
                Hello Time     2 sec     Max Age 20 sec     Forward Delay 15 sec
    Bridge ID             Priority             32770 (priority 32768 sys-id-ext 2)
                Address             0009.e91a.bc40
                Hello Time     2 sec     Max Age 20 sec     Forward Delay 15 sec
Interface                     Role Sts Cost                          Prio.Nbr     Type
------------------     ---- --- ---------
--------     --------------------------------
Gi4/0/0                      Desg FWD 20000                       128.1537     P2p
```

In the following example, interface gigabitethernet 4/0/3 with a service instance that has an outer encapsulation VLAN ID of 2 and a bridge domain of 100 receives a new service:

```
Router# enable
Router# configure terminal
Router(config)# interface gigabitethernet 4/0/3
Router((config-if)# service instance 2 ethernet
Router((config-if-srv)# encap dot1q 2 second-dot1q 100
Router((config-if-srv)# bridge-domain 200
```

Now there are two service instances configured on interface gigabitethernet 4/0/3 and both of them have the same outer VLAN 2.

```
interface GigabitEthernet4/0/3
        no ip address
    service instance 1 ethernet
    encapsulation dot1q 2
    bridge-domain 100
!
service instance 2 ethernet
    encapsulation dot1q 2 second-dot1q 100
        bridge-domain 200
```

The preceding configuration does not affect the MSTP operation on the interface; there is no state change for interface gi4/0/3 in the MST instance it belongs to.

```
Router# show spanning-tree mst 1
##### MST1     vlans mapped:            2
Bridge             address 0009.e91a.bc40         priority
32769 (32768 sysid 1)
Root             this switch for MST1
Interface                     Role Sts Cost                      Prio.Nbr Type
----------------                      ---- --- ---------     --------
--------------------------------
Gi4/0/3                    Desg FWD 20000                     128.1540 P2p
```

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 Commands List, All Releases

Standards

Standard	Title
MEF 6.1	Metro Ethernet Services Definitions Phase 2 (PDF 6/08)
MEF 10.1	Ethernet Services Attributes Phase 2 (PDF 10/06)

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 IEEE 802.1s on Bridge Domains

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 9 Feature Information for IEEE 802.1s on Bridge Domains

Feature Name	Releases	Feature Information
IEEE 802.1s on Bridge Domains	12.2(33)SRD 12.2(50)SY	The IEEE 802.1s on Bridge Domains feature enables MST on EVC interfaces. The following commands were introduced or modified: **bridge-domain (**service instance**), debug ethernet l2ctrl, debug l2ctrl**.

Configuring MAC Address Limiting on Service Instances Bridge Domains and EVC Port Channels

The MAC Address Limiting on Service Instances, Bridge Domains, and EVC Port Channels feature addresses port security with service instances by providing the capability to control and filter MAC address learning behavior at the granularity of a per-service instance. When a violation requires a shutdown, only the customer who is assigned to a given service instance is affected and--not all customers who are using the port. The MAC Address Security on EVC Port Channel feature supports MultiPoint Bridging over Ethernet (MPBE). MAC address limiting is a type of MAC security and is also referred to as a MAC security component or element.

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 MAC Address Limiting on Service Instances Bridge Domains and EVC Port Channels

- An understanding of service instances and bridge domains.
- An understanding of the concepts of MAC address limiting and how it is used for MAC security.
- An understanding of how port channels and EtherChannels work in a network.

Restrictions for MAC Address Limiting on Service Instances Bridge Domains and EVC Port Channels

MAC address limiting for service instances and bridge domains is configured under a service instance and is permitted only after the service instance is configured under a bridge domain. If a service instance is removed from a bridge domain, all the MAC address limiting commands under it are also removed. If a bridge domain is removed from a service instance, all the MAC address limiting commands are also removed.

Information About MAC Address Limiting on Service Instances Bridge Domains and EVC Port Channels

Ethernet Virtual Circuits Service Instances and Bridge Domains

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. It is an end-to-end representation of a single instance of a Layer 2 service being offered by a provider to a customer. An EVC embodies the different parameters on which the service is being offered. A service instance is the instantiation of an EVC on a given port.

Support for Ethernet bridging is an important Layer 2 service that is offered on a router as part of an EVC. Ethernet bridging enables the association of a bridge domain with a service instance.

Service instances are configured under a port channel. The traffic carried by service instances is load-balanced across member links. Service instances under a port channel are grouped and each group is associated with one member link. Ingress traffic for a single service instance can arrive on any member of

the bundle. All egress traffic for a service instance uses only one of the member links. Load-balancing is achieved by grouping service instances and assigning them to a member link.

For information about the Metro Ethernet Forum standards, see the Standards table in the Addition References section.

EVCs on Port Channels

An EtherChannel bundles individual Ethernet links into a single logical link that provides the aggregate bandwidth of up to eight physical links. The Ethernet Virtual Connection Services (EVCS) EtherChannel feature provides support for EtherChannels on service instances.

Note The MAC Address Security on EVC Port Channel services is supported only on bridge domains over Ethernet and is not supported on local connect and xconnect services.

EVCS uses the concepts of EVCs and service instances.

Load balancing is done on an Ethernet flow point (EFP) basis where a number of EFPs exclusively pass traffic through member links.

MAC Security and MAC Addressing

MAC security is enabled on a service instance by configuring the **mac security** command. Various MAC security elements can be configured or removed regardless of whether the **mac security** command is presently configured, but these configurations become operational only when the **mac security** command is applied.

In this document, the term “secured service instance” is used to describe a service instance on which MAC security is configured. The MAC addresses on a service instance on which MAC security is configured are referred to as “secured MAC addresses.” Secured MAC addresses can be either statically configured (as a permit list) or dynamically learned.

MAC Address Permit List

A permit list is a set of MAC addresses that are permitted on a service instance. Permitted addresses permanently configured into the MAC address table of the service instance.

On a service instance that is a member of a bridge domain, the operator is permitted to configure one or more permitted MAC addresses. For each permitted address, eligibility tests are performed and after the address passes these tests, it is either:

- Programmed into the MAC address table of the bridge domain, if MAC security is enabled on the service instance or.
- Stored in an area of memory referred to as “MAC table cache” if MAC security is not enabled on the service instance. When MAC security is enabled, the addresses from the MAC table cache are added to the MAC address table as secure addresses.

The eligibility tests performed when a user tries to add a MAC address to the permit list on a service instance are as follows:

1. If the address is already a denied address on the service instance, the configuration is rejected with an appropriate error message.
2. If the acceptance of this address would increase the secure address count on the service instance beyond the maximum number allowed, an attempt is made to make room by removing an existing address from

the MAC address table. The only candidate for removal is a dynamically learned address on the service instance. If sufficient room cannot be made, the configuration is rejected. If the acceptance of this address would increase the secure address count on the bridge domain beyond the maximum number allowed, an attempt is made to make room by removing an existing address from the MAC address table. The only candidate for removal is a dynamically learned address on the service instance. If room cannot be made, the configuration is rejected.

3 If the address is already permitted on another service instance in the same bridge domain, 0ne of the following actions occur:

   a If the conflicting service instance has MAC security configured, the configuration is rejected with an appropriate error message.
   b If the conflicting service instance does not have MAC security configured, the configuration is accepted silently. (If the operator attempts to enable MAC security on the conflicting service instance, that attempt fails.)

MAC Address Deny List

A deny list is a set of MAC addresses that are not permitted on a service instance. An attempt to learn a denied MAC address will fail. On a service instance that is a member of a bridge domain, the operator is permitted to configure one or more denied MAC addresses. The arrival of a frame with a source MAC address that is part of a deny list will trigger a violation response.

Before a denied address can be configured, the following test is performed:

1 If the address is already configured as a permitted address on the specific service instance or if the address has been learned and saved as a sticky address on the service instance, the configuration is rejected with an appropriate error message.

In all other cases, the configuration of the denied address is accepted. Typical cases include:

- The address is configured as a permitted address on another service instance in the same bridge domain, or the address has been learned and saved as a sticky address on another service instance.
- The address is present in the MAC table of the bridge domain as a dynamically learned address on the specific service instance and is deleted from the MAC table before the configuration is accepted.

MAC Address Limiting and Learning

An upper limit for the number of secured MAC addresses allowed on a bridge domain service instance can be configured. This limit includes addresses added as part of a permit list and dynamically learned MAC addresses.

Before an unknown MAC address is learned, a series of checks are run against a set of configured and operational constraints. If any of these checks fails, the address is not learned, and a configured violation response is triggered.

- Static and Dynamic MAC Addresses, page 319
- Dynamic MAC Address Learning, page 319
- MAC Address Limiting on Service Instances, page 319
- MAC Address Limiting for Bridge Domains, page 319
- Relationship Between the MAC Address Limit on a Bridge Domain and on a Service Instance, page 319
- MAC Move and MAC Locking, page 320

Static and Dynamic MAC Addresses

A static MAC address is specified as permitted on a service instance, by a **mac security permit**command. A dynamic MAC address is a source MAC address encountered by the service instance that is not present in the MAC table but is allowed into and learned by the MAC address table.

Dynamic MAC Address Learning

Dynamic MAC address learning occurs play when the bridging data path encounters an ingress frame whose source address is not present in the MAC address table for the ingress secured service instance.

The MAC security component is responsible for permitting or denying the addition of the new source address into the MAC table. The following constraints apply:

1. In considering if this MAC address is to be learned, a check to see whether the number of secured MAC addresses will exceed the maximum number that are permitted to be learned on the individual service instance and on the bridge domain as a whole, or not is performed.
2. A check is performed to determine if the MAC address now being seen on another service instance was learned previously on a secured service instance in the same bridge domain.
3. A check is performed to verify that the new dynamic MAC address is a deny list.

MAC Address Limiting on Service Instances

The user can configure the maximum number of MAC addresses that can exist in the MAC table that is associated with a service instance. This number includes statically configured and dynamically learned (including sticky) addresses.

On a service instance that has MAC security enabled and that does not have the maximum number of MAC addresses configured, the number of addresses allowed is one. This means that if the service instance has an associated permit list, that permit list can have only one address, and no addresses are learned dynamically. If the service instance does not have an associated permit list, one MAC address may be learned dynamically.

MAC Address Limiting for Bridge Domains

An upper limit for the number of MAC addresses that can reside in the MAC address table of a bridge domain can be set. This is set independently of the upper limit of secured MAC addresses on the service instance. An attempted violation of this bridge domain MAC address limit will cause the MAC address learn attempt to fail, and the frame to be dropped.

If the bridge domain MAC address limit is not configured, then by default, the maximum number of MAC addresses allowed on a bridge domain is the maximum number that can be supported by that platform.

Relationship Between the MAC Address Limit on a Bridge Domain and on a Service Instance

The MAC security commands permit the user to specify the maximum count of MAC table entries on a bridge domain and on a service instance simultaneously. However, there are no restrictions on the count that is configured on the service instance.

The table below shows an example of an initial configuration where three service instances are configured on a bridge domain:

Table 10 Bridge-Domain and Service-Instance MAC Address Limit

Bridge-Domain / Service-Instance Number	MAC Address Limit
Bridge Domain 1000	20
Service Instance 1001	5
Service Instance 1002	10
Service Instance 1003	To be configured

If the user wishes to configure MAC security on service instance 1003, any value can be configured for the maximum count. For example:

```
service instance 1003 ethernet
 bridge-domain 1
 mac security
 mac security maximum addresses 35
```

A MAC address limit of 35 is permitted, even though the total MAC address limit for the three service instances (5 + 10 + 35) would exceed the count (20) configured on the bridge domain. Note that during actual operation, the bridge domain limit of 20 is in effect. The dynamic secure address count cannot exceed the lowest count applicable, so it is not possible for service instance 1003 to learn 35 addresses.

MAC Move and MAC Locking

If a MAC address is present in the MAC address table for a service instance (for example, service instance 1) on which MAC security is configured, the same MAC address cannot be learned on another service instance (for example, service instance 2) in the same bridge domain.

If service instance 2 attempts to learn the same MAC address, the violation response configured on service instance 2 is triggered. If MAC security is not configured on service instance 2 and a violation response is not configured, the “shutdown” response sequence is triggered on service instance 2.

If MAC security is not enabled on service instance 1, the violation is not triggered. service instance 2 learns the MAC address and moves it from service instance 1.

For some platforms such as Cisco 7600 series routers, MAC address moves are allowed but moves between secured service instances and nonsecured service instances cannot be detected.

For example, if you do not configure MAC security on service instance 2 because of a hardware limitation on the Cisco 7600 series router, a MAC move from secured service instance 1 to service instance 2 is accepted. Therefore, it is recommended that all service instances within the same bridge-domain are configured as secured service instances.

Violation Response Configuration

A violation response is a response to a MAC security violation or a failed attempt to dynamically learn a MAC address due to an address violation. MAC security violations are of two types:

Type 1 Violation --The address of the ingress frame cannot be dynamically learned due to a deny list, or because doing so would cause the maximum number of secure addresses to be exceeded (see the MAC Address Limiting and Learning, page 318).

Type 2 Violation --The address of the ingress frame cannot be dynamically learned because it is already “present” on another secured service instance (see the MAC Move and MAC Locking, page 320).

There are three possible sets of actions that can be taken in response to a violation:

1 **Shutdown**
2 The ingress frame is dropped.
3 The service instance on which the offending frame arrived is shut down.
4 The violation count is incremented, and the violating address is recorded for later CLI display.
5 The event and the response are logged to SYSLOG.
6 **Restrict**
7 The ingress frame is dropped.
8 The violation count is incremented, and the violating address is recorded for display.
9 The event and the response are logged to SYSLOG.
10 **Protect**
11 The ingress frame is dropped.

If a violation response is not configured, the default response mode is shutdown. The violation response can be configured to protect or restrict mode. A “no” form of a violation response, sets the violation response to the default mode of shutdown.

You are allowed to configure the desired response for a Type 1 and Type 2 violations on a service instance. For a Type 1 violation on a bridge domain (that is, if the learn attempt conforms to the policy configured on the service instance, but violates the policy configured on the bridge domain), the response is always “Protect.” This is not configurable.

In shutdown mode, the service instance is put into the error disabled state immediate, an SNMP trap notification is transmitted, and a message is sent to the console and SYSLOG as shown below:

```
%ETHER_SERVICE-6-ERR_DISABLED:
Mac security violation - shutdown service instance 100 on interface gig 0/0/0
```

To bring a service instance out of the error-disabled state, use **errdisable recovery cause mac-security**command to set the auto recovery timer, or re-enable it using the EXEC command **clear ethernet service instance id** *id* **interface** *type number* **errdisable**.

In Restrict mode, the violation report is sent to SYSLOG at level LOG_WARNING.

Support for the different types of violation responses depends on the capabilities of the platform. The desired violation response can be configured on the service instance. The configured violation response does not take effect unless and until MAC security is enabled using the **mac security** command.

MAC Address Aging Configuration

A specific time scheduler can be set to age out secured MAC addresses that are dynamically learned or statically configured on both service instances and bridge domains, thus freeing up unused addresses from the MAC address table for other active subscribers.

The set of rules applied to age out secured MAC addresses is called secure aging. By default, the entries in the MAC address table of a secured service instance are never aged out. This includes permitted addresses and dynamically learned addresses.

The **mac security aging time** *aging-time* command sets the aging time of the addresses in the MAC address table to <*n* > minutes. By default, this affects only dynamically learned (not including sticky) addresses--permitted addresses and sticky addresses are not affected by the application of this command.

By default, the aging time <n> configured via the **mac security aging time** *aging-time* command is an absolute time. That is, the age of the MAC address is measured from the instant that it was first encountered on the service instance. This interpretation can be modified by using the **mac security aging**

time *aging-time* **inactivity** command, which specifies that the age <n> be measured from the instant that the MAC address was last encountered on the service instance.

The **mac security aging static**and **mac security aging sticky** commands specify that the **mac security aging time**aging-time command must be applicable to permitted and sticky MAC addresses, respectively. In the case of permitted MAC addresses, the absolute aging time is measured from the time the address is entered into the MAC address table (for example, when it is configured or whenever the **mac security** command is entered--whichever is later).

If the **mac security aging time** command is not configured, the **mac security aging static** command has no effect.

Sticky MAC Address Configurations

The ability to make dynamically learned MAC addresses on secured service instances permanent even after interface transitions or device reloads can be set up and configured. A dynamically learned MAC address that is made permanent on a secured service instance is called a “sticky MAC address”. The **mac security sticky** command is used to enable the sticky MAC addressing feature on a service instance.

With the “sticky” feature enabled on a secured service instance, MAC addresses learned dynamically on the service instance are kept persistent across service instance line transitions and device reloads.

The sticky feature has no effect on statically configured MAC addresses. The sticky addresses are saved in the running configuration. Before the device is reloaded, it is the responsibility of the user to save the running configuration to the startup configuration. Doing this will ensure that when the device comes on, all the MAC addresses learned dynamically previously are immediately populated into the MAC address table.

The **mac security sticky address** *mac-address* command can configure a specific MAC address as a sticky MAC address. The use of this command is not recommended for the user because configuring a MAC address as a static address does the same thing. When sticky MAC addressing is enabled by the **mac security sticky** command, the dynamically learned addresses are marked as sticky and a **mac security sticky address** *mac-address* command is automatically generated and saved in the running configuration for each learned MAC address on the service instances.

- Aging for Sticky Addresses, page 322

Aging for Sticky Addresses

MAC addresses learned on a service instance that has the sticky behavior enabled are subject to aging as configured by the **mac security aging time** and **mac security aging sticky** commands. In other words, for the purpose of aging functionality, sticky addresses are treated the same as dynamically learned addresses.

Transitions

This section contains a description of the expected behavior of the different MAC security elements when various triggers are applied; for example, configuration changes or link state transitions.

- MAC Security Enabled on a Service Instance, page 323
- MAC Security Disabled on a Service Instance, page 323
- Service Instance Moved to a New Bridge Domain, page 323
- Service Instance Removed from a Bridge Domain, page 323
- Service Instance Shut Down Due to Violation, page 323
- Interface Service Instance Down Linecard OIR Removed, page 323

MAC Security Enabled on a Service Instance

When MAC security is enabled on a service instance, all existing MAC table entries for the service instance are purged. Then, permitted MAC address entries and sticky addresses are added to the MAC table, subject to the prevailing MAC address limiting constraints on the bridge domain.

If MAC address limits are exceeded, any MAC address that fails to get added is reported via an error message to the console, the attempt to enable MAC security on the service instance fails, and the already added permitted entries are backed out or removed.

The aging timer for all entries is updated according to the secure aging rules.

MAC Security Disabled on a Service Instance

The existing MAC address table entries for this service instance are purged.

Service Instance Moved to a New Bridge Domain

This transition sequence applies to all service instances, whether or not they have MAC security configured. All the MAC addresses on this service instance in the MAC address table of the old bridge domain are removed. The count of dynamically learned addresses in the old bridge domain is decremented. Then, all the MAC security commands are permanently erased from the service instance.

Service Instance Removed from a Bridge Domain

All the MAC addresses in the MAC address table that attributable to this service instance are removed, and the count of dynamically learned addresses in the bridge domain is decremented. Since MAC security is applicable only on service instances that are members of a bridge domain, removing a service instance from a bridge domain causes all the MAC security commands to be erased permanently.

Service Instance Shut Down Due to Violation

All dynamically learned MAC addresses in the MAC address table are removed, and all the other MAC security state values are left unchanged. The only change is that no traffic is forwarded, and therefore no learning can take place.

Interface Service Instance Down Linecard OIR Removed

The MAC tables of all the affected bridge domains are cleared of all the entries attributable to the service instances that are down.

Interface Service Instance Re-activated Linecard OIR Inserted

The static and sticky address entries in the MAC tables of the affected bridge domains are re-created to the service instances that are activated.

MAC Address Limit Decreased

When the value of the MAC address limit on the service instance is changed initially, a sanity check is performed to ensure that the new value of <n> is greater than or equal to the number of permitted entries. If not, the command is rejected. The MAC table is scanned for addresses that are attributable to this service instance, and dynamically learned MAC addresses are removed when the new MAC address limit is less than the old MAC address limit.

When the value of <n> on a bridge domain is changed initially, a sanity check is performed to ensure that the new value of <n> is greater than or equal to the sum of the number of permitted entries on all the secured service instances on the bridge domain. If this sanity test fails, the command is rejected. The bridge domain MAC address table (regardless of service instance) is scanned for dynamically learned (or sticky) addresses. All dynamically learned addresses are removed when the new MAC address limit is less than the old MAC address limit.

Sticky Addresses Added or Removed on a Service Instance

Existing dynamically learned MAC addresses remain unchanged. All new addresses learned become “sticky” addresses.

Disabling sticky addresses causes all sticky secure MAC addresses on the service instance to be removed from the MAC address table. All new addresses learned become dynamic addresses on the service instance and are subject to aging.

How to Configure MAC Address Limiting on Service Instances Bridge Domains and EVC Port Channels

Enabling MAC Security on a Service Instance

Perform this task to enable MAC address security on a service instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot/subslot/port*
4. **service instance** *id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **bridge-domain** *bridge-id*
7. **mac security**
8. **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 gigabitethernet** *slot/subslot/port* **Example:** `Router(config)# interface gigabitethernet 1/0/0`	Specifies the type and location of the interface to configure.

	Command or Action	Purpose
Step 4	**service instance** *id* **ethernet** **Example:** `Router(config-if)# service instance 100 ethernet`	Creates a service instance on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.
Step 6	**bridge-domain** *bridge-id* **Example:** `Router(config-if-srv)# bridge-domain 200`	Binds the service instance to a bridge- domain instance where *bridge-id*is the identifier for the bridge- domain instance.
Step 7	**mac security** **Example:** `Router(config-if-srv)# mac security`	Enables MAC security on the service instance.
Step 8	**end** **Example:** `Router(config-if-srv)# end`	Exits service instance configuration mode and and enters privileged EXEC mode.

Enabling MAC Security on an EVC Port Channel

Note

- All member links of the port channel are on Cisco 7600-ES+ line cards.
- Bridge- domain, xconnect, connect EVCs, switchports, and IP subinterfaces are allowed over the port channel interface and the main interface.
- If you configure a physical port as part of a channel group, you cannot configure EVCs under that physical port.
- A physical port that is part of an EVC port channel cannot have switchport configuration.
- Statically configuring port channel membership with Link Aggregation Control Protocol (LACP) is not supported.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface port-channel** *channel-group*
4. **service instance** *id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **bridge-domain** *bridge-id*
7. **mac security**
8. 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 port-channel** *channel-group* **Example:** `Router(config)# interface port-channel 2`	Specifies the port channel group number and enters interface configuration mode. • Acceptable values are integers from 1 to 64.
Step 4	**service instance** *id* **ethernet** **Example:** `Router(config-if)# service instance 100 ethernet`	Creates a service instance on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.

	Command or Action	Purpose
Step 6	**bridge-domain** *bridge-id* **Example:** `Router(config-if-srv)# bridge-domain 200`	Binds the service instance to a bridge- domain instance where *bridge-id*is the identifier for the bridge- domain instance.
Step 7	**mac security** **Example:** `Router(config-if-srv)# mac security`	Enables MAC security on the service instance.
Step 8	end **Example:** `Router(config-if-srv)# end`	Exits service instance configuration mode and and enters privileged EXEC mode.

Configuring a MAC Address Permit List

Perform this task to configure permitted MAC addresses on a service instance that is a member of a bridge domain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot /subslot/port*
4. **service instance** *id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **bridge-domain** *bridge-id*
7. **mac security address permit** *mac-address*
8. **mac security address permit** *mac-address*
9. **mac security address permit** *mac-address*
10. **mac security address permit** *mac-address*
11. **mac security address permit** *mac-address*
12. **mac security**
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	**interface gigabitethernet** *slot /subslot/port* **Example:** `Router(config)# interface gigabitethernet 2/0/1`	Specifies the type and location of the interface to configure, where: • *type* --Specifies the type of the interface. • *number* --Specifies the location of the interface.
Step 4	**service instance** *id* **ethernet** **Example:** `Router(config-if)# service instance 100 ethernet`	Creates a service instance (an instance of an EVC) on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to be used for mapping ingress dot1q frames on an interface to the appropriate service instance.
Step 6	**bridge-domain** *bridge-id* **Example:** `Router(config-if-srv)# bridge-domain 200`	Binds the service instance to a bridge- domain instance where *bridge-id* is the identifier for the bridge- domain instance.
Step 7	**mac security address permit** *mac-address* **Example:** `Router(config-if-srv)# mac security address permit a2aa.aaaa.aaaa`	Adds the specified MAC address as a permit MAC address for the service instance.

	Command or Action	Purpose
Step 8	**mac security address permit** *mac-address* **Example:** `Router(config-if-srv)# mac security address permit a2aa.aaaa.aaab`	Adds the specified MAC address as a permitted MAC address for the service instance.
Step 9	**mac security address permit** *mac-address* **Example:** `Router(config-if-srv)# mac security address permit a2aa.aaaa.aaac`	Adds the specified MAC address as a permitted MAC address for the service instance.
Step 10	**mac security address permit** *mac-address* **Example:** `Router(config-if-srv)# mac security address permit a2aa.aaaa.aaad`	Adds the specified MAC address as a permitted MAC address for the service instance.
Step 11	**mac security address permit** *mac-address* **Example:** `Router(config-if-srv)# mac security address permit a2aa.aaaa.aaae`	Adds the specified MAC address as a permitted MAC address for the service instance.
Step 12	**mac security** **Example:** `Router(config-if-srv)# mac security`	Enables MAC security on the service instance.
Step 13	**end** **Example:** `Router(config-if-srv)# end`	Exits service instance configuration mode and and enters privileged EXEC mode.

Configuring a MAC Address Deny List

Perform this task to configure a list of MAC addresses that are not allowed on a service instance that is a member of a bridge domain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot /subslot/port*
4. **service instance** *id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **bridge-domain** *bridge-id*
7. **mac security address deny** *mac-address*
8. **mac security address deny** *mac-address*
9. **mac security address deny** *mac-address*
10. **mac security address deny** *mac-address*
11. **mac security address deny** *mac-address*
12. **mac security**
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	**interface gigabitethernet** *slot /subslot/port* **Example:** `Router(config)# interface gigabitethernet 1/0/1`	Specifies the type and location of the interface to configure, where: • *type* --Specifies the type of the interface. • *number* --Specifies the location of the interface.
Step 4	**service instance** *id* **ethernet** **Example:** `Router(config-if)# service instance 100 ethernet`	Creates a service instance (an instance of an EVC) on an interface and enters service instance configuration mode.

	Command or Action	Purpose
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.
Step 6	**bridge-domain** *bridge-id* **Example:** `Router(config-if-srv)# bridge-domain 200`	Binds the service instance to a bridge- domain instance where *bridge-id* is the identifier for the bridge- domain instance.
Step 7	**mac security address deny** *mac-address* **Example:** `Router(config-if-srv)# mac security address deny a2aa.aaaa.aaaa`	Adds the specified MAC address as a denied MAC address for the service instance.
Step 8	**mac security address deny** *mac-address* **Example:** `Router(config-if-srv)# mac security address deny a2aa.aaaa.aaab`	Adds the specified MAC address as a denied MAC address for the service instance.
Step 9	**mac security address deny** *mac-address* **Example:** `Router(config-if-srv)# mac security address deny a2aa.aaaa.aaac`	Adds the specified MAC address as a denied MAC address for the service instance.
Step 10	**mac security address deny** *mac-address* **Example:** `Router(config-if-srv)# mac security address deny a2aa.aaaa.aaad`	Adds the specified MAC address as a denied MAC address for the service instance.
Step 11	**mac security address deny** *mac-address* **Example:** `Router(config-if-srv)# mac security address deny a2aa.aaaa.aaae`	Adds the specified MAC address as a denied MAC address for the service instance.

	Command or Action	Purpose
Step 12	**mac security** **Example:** `Router(config-if-srv)# mac security`	Enables MAC security on the service instance.
Step 13	**end** **Example:** `Router(config-if-srv)# end`	Exits service instance configuration mode and and enters privileged EXEC mode.

Configuring MAC Address Limiting on a Bridge Domain

Perform this task to configure an upper limit for the number of secured MAC addresses that reside in a bridge domain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **bridge-domain** *bridge-id*
4. **mac limit maximum addresses** *maximum-addresses*
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.

	Command or Action	Purpose
Step 3	**bridge-domain** *bridge-id* **Example:** `Router(config)# bridge-domain 100`	Configures components on a bridge domain and enters bridge-domain configuration mode.
Step 4	**mac limit maximum addresses** *maximum-addresses* **Example:** `Router(config-bdomain)# mac limit maximum addresses 200`	Sets the MAC limit maximum addresses.
Step 5	end **Example:** `Router(config-bdomain)# end`	Exits bridge domain configuration mode and enters privileged EXEC mode.

ConfiguringMACAddressLimitingonaServiceInstance

Perform this task to configure an upper limit for the number of secured MAC addresses allowed on a service instance. This number includes addresses added as part of a permit list as well as dynamically learned MAC addresses. If the upper limit is decreased, all learned MAC entries are removed.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance** *id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **bridge-domain** *bridge-id*
7. **mac security maximum addresses maximum-addresses**
8. **mac security**
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 gigabitethernet 2/0/1`	Specifies the type and location of the interface to configure, where: • *type* --Specifies the type of the interface. • *number* --Specifies the location of the interface.
Step 4	**service instance** *id* **ethernet** **Example:** `Router(config-if)# service instance 100 ethernet`	Creates a service instance (an instance of an EVC) on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to be used to map ingress dot1q frames on an interface to the appropriate service instance.
Step 6	**bridge-domain** *bridge-id* **Example:** `Router(config-if-srv)# bridge-domain 200`	Binds the service instance to a bridge- domain instance where *bridge-id* is the identifier for the bridge- domain instance.
Step 7	**mac security maximum addresses maximum-addresses** **Example:** `Router(config-if-srv)# mac security maximum addresses 500`	Sets the maximum number of secure addresses permitted on the service instance.

	Command or Action	Purpose
Step 8	**mac security** **Example:** `Router(config-if-srv)# mac security`	Enables MAC security on the service instance.
Step 9	**end** **Example:** `Router(config-if-srv)# end`	Exits service instance configuration mode and enters privileged EXEC mode.

Configuring a MAC Address Violation

Perform this task to specify the expected behavior of a device when an attempt to dynamically learn a MAC address fails because the configured MAC security policy on the service instance was violated.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot /subslot/port*
4. **service instance** *id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **bridge-domain bridge-id**
7. Do one of the following:
   - **mac security violation restrict**
   - **mac security violation protect**
8. **mac security**
9. **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 gigabitethernet** *slot /subslot/port* **Example:** **Example:** `Router(config)# interface gigabitethernet 2/0/1`	Specifies the type and location of the interface to configure, where: • *type* --Specifies the type of the interface. • *number* --Specifies the location of the interface.
Step 4	**service instance** *id* **ethernet** **Example:** `Router(config-if)# service instance 100 ethernet`	Creates a service instance (an instance of an EVC) on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to be used to map ingress dot1q frames on an interface to the appropriate service instance.
Step 6	**bridge-domain bridge-id** **Example:** `Router(config-if-srv)# bridge-domain 100`	Binds the service instance to a bridge- domain instance where *bridge-id* is the identifier for the bridge- domain instance.

	Command or Action	Purpose

Step 7	Do one of the following:
	• **mac security violation restrict**
	• **mac security violation protect**
	Example:
	`Router(config-if-srv)# mac security violation restrict`
	Example:
	`Router(config-if-srv)# mac security violation protect`
		Sets the violation mode (for Type 1 and 2 violations) to restrict.
		or
		Sets the violation mode (for Type 1 and 2 violations) to protect.
		• If a MAC security violation response is not specified, by default, the violation mode is shutdown.

Step 8	**mac security** **Example:** `Router(config-if-srv)# mac security`	Enables MAC security on the service instance.
Step 9	**end** **Example:** `Router(config-if-srv)# end`	Exits service instance configuration mode and enters privileged EXEC mode.

Configuring MAC Address Aging

Perform this task to configure the aging of secured MAC addresses under MAC security. Secured MAC addresses are not subject to the normal aging of MAC table entries. If aging is not configured, secured MAC addresses are never aged out.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* /*subslot*/*port*
4. **service instance** *id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **bridge-domain** *bridge-id*
7. **mac security aging time** *aging-time* **[inactivity]**
8. **mac security**
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 gigabitethernet** *slot /subslot/port* **Example:** **Example:** `Router(config)# interface gigabitethernet 2/0/1`	Specifies the type and location of the interface to configure, where: • *type* --Specifies the type of the interface. • *number* --Specifies the location of the interface.
Step 4	**service instance** *id* **ethernet** **Example:** `Router(config-if)# service instance 100 ethernet`	Creates a service instance (an instance of an EVC) on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.
Step 6	**bridge-domain** *bridge-id* **Example:** `Router(config-if-srv)# bridge-domain 200`	Binds the service instance to a bridge- domain instance where *bridge-id* is the identifier for the bridge- domain instance.

	Command or Action	Purpose
Step 7	**mac security aging time** *aging-time* **[inactivity]** **Example:** `Router(config-if-srv)# mac security aging time 200 inactivity`	Sets the aging time for secure addresses, in minutes. The optional **inactivity** keyword specifies that the aging out of addresses is based on inactivity of the sending hosts (as opposed to absolute aging).
Step 8	**mac security** **Example:** `Router(config-if-srv)# mac security`	Enables MAC security on the service instance.
Step 9	**end** **Example:** `Router(config-if-srv)# end`	Exits service instance configuration mode and enters privileged EXEC mode.

Configuring a Sticky MAC Address

If sticky MAC addressing is configured on a secured service instance, MAC addresses that are learned dynamically on the service instance are retained during a link-down condition. Perform this task to configure sticky MAC addresses on a service instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot /subslot/port*
4. **service instance** *id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **bridge-domain** *bridge-id*
7. **mac security sticky**
8. **mac security**
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 gigabitethernet** *slot /subslot/port* **Example:** **Example:** `Router(config)# interface gigabitethernet 2/0/1`	Specifies the type and location of the interface to configure, where: • *type* --Specifies the type of the interface. • *number* --Specifies the location of the interface.
Step 4	**service instance** *id* **ethernet** **Example:** `Router(config-if)# service instance 100 ethernet`	Creates a service instance (an instance of an EVC) on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to be used to map ingress dot1q frames on an interface to the appropriate service instance.
Step 6	**bridge-domain** *bridge-id* **Example:** `Router(config-if-srv)# bridge-domain 200`	Binds the service instance to a bridge- domain instance where *bridge-id* is the identifier for the bridge- domain instance.

	Command or Action	Purpose
Step 7	**mac security sticky** **Example:** `Router(config-if-srv)# mac security sticky`	Enables sticky behavior on the service instance.
Step 8	**mac security** **Example:** `Router(config-if-srv)# mac security`	Enables MAC security on the service instance.
Step 9	**end** **Example:** `Router(config-if-srv)# end`	Exits service instance configuration mode and enters privileged EXEC mode.

Displaying the MAC Security Status of a Specific Service Instance

Perform this task to display the MAC security status of a service instance.

SUMMARY STEPS

1. **enable**
2. **show ethernet service instance id** *id* **interface** *type number* **mac security**

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 service instance id** *id* **interface** *type number* **mac security** **Example:** `Router#` **show ethernet service instance id 100 interface GigabitEthernet 1/1 mac security**	Displays the MAC security status of a specific service instance.

Displaying the Service Instances with MAC Security Enabled

Perform this task to display all the service instances with MAC security enabled.

SUMMARY STEPS

1. **enable**
2. **show ethernet service instance mac security**

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 service instance mac security** **Example:** `Router# show ethernet service instance mac security`	Displays all the service instances with MAC security enabled.

Displaying the Service Instances with MAC Security Enabled on a Specific Bridge Domain

Perform this task to display the service instances on a specific bridge domain that have MAC security enabled.

SUMMARY STEPS

1. **enable**
2. **show bridge-domain** *id* **mac security**

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	**show bridge-domain** *id* **mac security** **Example:** `Router#` **show bridge-domain 100 mac security**	Displays all the service instances with MAC security enabled on a specific bridge domain.

Showing the MAC Addresses of All Secured Service Instances

Perform this task to display all the MAC addresses on all the secured service instances.

Note For some platforms such as Cisco 7600 series routers, the MAC address remaining age time information is available only on the switch console. Use the **remote command switch**command and the **show ethernet service instance mac security address** command to display the MAC address remaining age time information.

>

SUMMARY STEPS

1. **enable**
2. **show ethernet service instance mac security address**

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 service instance mac security address** **Example:** `Router#` **show ethernet service instance mac security address**	Displays the secured addresses on all the service instances.

Showing the MAC Addresses of a Specific Service Instance

Perform this task to display all the MAC addresses of a specific service instance.

Note For some platforms such as Cisco 7600 routers, the MAC address remaining age time information is available only on the switch console. Use the **remote command switch**command and the **show ethernet service instance id** *id* **interface** *type number* **mac security address** command to display the MAC address remaining age time information.

>

SUMMARY STEPS

1. **enable**
2. **show ethernet service instance id** *id* **interface** *type number* **mac security address**

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 service instance id** *id* **interface** *type number* **mac security address** **Example:** `Router# show ethernet service instance id 200 interface GigabitEthernet 1/0 mac security address`	Displays the addresses of a specific service instance.

Showing the MAC Addresses of All Service Instances on a Specific Bridge Domain

Perform this task to display the MAC addresses of all service instances on a specific bridge domain.

Note For some platforms such as Cisco 7600 series routers, the MAC address remaining age time information is available only on the switch console. Use the **remote command switch**command and the **show bridge-domain id mac security address** command to display the MAC address remaining age time information.

>

SUMMARY STEPS

1. **enable**
2. **show bridge-domain** *id* **mac security address**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show bridge-domain** *id* **mac security address** **Example:** `Router#` **show bridge-domain 100 mac security address**	Displays the secured addresses of all the service instances on a specified bridge domain.

Showing the MAC Security Statistics of a Specific Service Instance

This section describes how to display the MAC security statistics of a specific service instance.

SUMMARY STEPS

1. **enable**
2. **show ethernet service instance id** *id* **interface** *type number* **mac security statistics**

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 service instance id** *id* **interface** *type number* **mac security statistics** **Example:** `Router#` **show ethernet service instance id 100 interface GigabitEthernet 1/1 mac security statistics**	Displays the MAC security statistics of a specific service instance.

Showing the MAC Security Statistics of All Service Instances on a Specific Bridge Domain

Perform this task to display the MAC security statistics of all the service instances on a specific bridge domain.

SUMMARY STEPS

1. **enable**
2. **show bridge-domain** *bridge-id* **mac security statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show bridge-domain** *bridge-id* **mac security statistics** **Example:** `Router#` **show bridge-domain 100 mac security statistics**	Displays the MAC security statistics of all service instances that belong to a specific bridge domain.

Showing the Last Violation Recorded on Each Service Instance on a Specific Bridge Domain

Perform this task to display the last violation recorded on each service instance on a specific bridge domain. Service instances on which there have been no violations are excluded from the output.

SUMMARY STEPS

1. **enable**
2. **show bridge-domain** *bridge-id* **mac security last violation**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show bridge-domain** *bridge-id* **mac security last violation** **Example:** `Router#` **show bridge-domain 100 mac security last violation**	Displays information about the last violation recorded on each of the service instances that belong to the bridge domain.

Clearing All Dynamically Learned MAC Addresses on a Service Instance

Perform this task to clear all dynamically learned MAC addresses on a service instance.

SUMMARY STEPS

1. **enable**
2. **clear ethernet service instance id** *id* **interface** *type number* **mac table**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ethernet service instance id** *id* **interface** *type number* **mac table** **Example:** `Router#` **clear ethernet service instance id 100 interface GigaBitEthernet 1/1 mac table**	Clears all the dynamically learned MAC addresses on the specified service instance.

Clearing All Dynamically Learned MAC Addresses on a Bridge Domain

Perform this task to clear all dynamically learned MAC addresses on a bridge domain.

SUMMARY STEPS

1. **enable**
2. **clear bridge-domain bridge-id mac table**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear bridge-domain bridge-id mac table** **Example:** `Router# clear bridge-domain 100 mac table`	Clears all dynamically learned MAC addresses on the specified bridge domain.

Bringing a Specific Service Instance Out of the Error-Disabled State

Perform this task to bring a specific service instance out of the error-disabled state.

Note The **clear ethernet service instance id** *id* **interface** type *number* **errdisable**command can also be used to bring a service instance out of an error disabled state. See the Bringing a Specific Service Instance Out of the Error-Disabled State, page 351.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance** *id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **bridge-domain** *bridge-id*
7. **mac security**
8. **errdisable recovery cause mac-security** *interval*
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:** **Example:** `Router(config)# interface gigabitethernet 2/0/1`	Specifies the type and location of the interface to configure, where: • *type* --Specifies the type of the interface. • *number* --Specifies the location of the interface.
Step 4	**service instance** *id* **ethernet** **Example:** `Router(config-if)# service instance 100 ethernet`	Creates a service instance (an instance of an EVC) on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Defines the matching criteria to be used to map ingress dot1q frames on an interface to the appropriate service instance.
Step 6	**bridge-domain** *bridge-id* **Example:** `Router(config-if-srv)# bridge-domain 200`	Binds the service instance to a bridge-domain instance where *bridge-id* is the identifier for the bridge-domain instance.

	Command or Action	Purpose
Step 7	**mac security** **Example:** `Router(config-if-srv)# mac security`	Enables MAC security on the service instance.
Step 8	**errdisable recovery cause mac-security** *interval* **Example:** `Router(config-if-srv)#` **errdisable recovery cause mac-security 50**	Brings a specific service instance out of an error-disabled state and specifies a time interval to recover.
Step 9	**end** **Example:** `Router(config-if-srv)# end`	Exits service instance configuration mode and enters privileged EXEC mode.

Bringing a Specific Service Instance Out of the Error-Disabled State

Perform this task to bring a specific service instance out of the error-disabled state.

SUMMARY STEPS

1. **enable**
2. **clear ethernet service instance id id interface type number errdisable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear ethernet service instance id id interface type number errdisable** **Example:** `Router#` **clear ethernet service instance id 100 interface FastEthernet 1/1 errdisable**	Brings a specific service instance out of the error-disabled state (shutdown).

Configuration Examples for MAC Address Limiting on Service Instances and Bridge Domains and EVC Port Channels

Example Enabling MAC Security on a Service Instance

The following example shows how to enable MAC security on a service instance:

```
Router> enable
Router# configure terminal
Router(config)# interface gigabitethernet 3/0/1
Router(config-if)# service instance 100 ethernet
Router(config-if-srv)# encapsulation dot1Q 100
Router(config-if-srv)# bridge-domain 100
Router(config-if-srv)# mac security
Router(config-if-srv)# end
```

Example Enabling MAC Security on an EVC Port Channel

The following example shows how to enable MAC Security on an EVC port channel:

```
Router> enable
Router# configure terminal
Router(config)# interface port-channel 2
Router(config-if)# service instance 100 ethernet
Router(config-if-srv)# encapsulation dot1Q 100
Router(config-if-srv)# bridge-domain 100
Router(config-if-srv)# mac security
Router(config-if-srv)# end
```

Example Configuring a MAC Address Permit List

The following example shows how to configure a MAC address permit list:

```
Router> enable
Router# configure terminal
Router(config)# interface gigabitethernet 3/0/1
Router(config-if)# service instance 100 ethernet
Router(config-if-srv)# encapsulation dot1Q 100
Router(config-if-srv)# bridge-domain 100
Router(config-if-srv)# mac security address permit a2aa.aaaa.aaaa
Router(config-if-srv)# mac security address permit a2aa.aaaa.aaab
Router(config-if-srv)# mac security address permit a2aa.aaaa.aaac
Router(config-if-srv)# mac security address permit a2aa.aaaa.aaad
Router(config-if-srv)# mac security address permit a2aa.aaaa.aaae
Router(config-if-srv)# mac security
Router(config-if-srv)# end
```

Example Configuring a MAC Address Deny List

The following example shows how to configure a MAC address deny list:

```
Router> enable
Router# configure terminal
Router(config)# interface gigabitethernet 3/0/1
Router(config-if)# service instance 100 ethernet
Router(config-if-srv)# encapsulation dot1Q 100
Router(config-if-srv)# bridge-domain 100
Router(config-if-srv)# mac security address deny a2aa.aaaa.aaaa
Router(config-if-srv)# mac security address deny a2aa.aaaa.aaab
Router(config-if-srv)# mac security address deny a2aa.aaaa.aaac
Router(config-if-srv)# mac security address deny a2aa.aaaa.aaad
Router(config-if-srv)# mac security address deny a2aa.aaaa.aaae
Router(config-if-srv)# mac security
Router(config-if-srv)# end
```

Example Configuring MAC Address Limiting on a Bridge Domain

```
Router> enable
Router# configure terminal
Router(config)# bridge-domain 100
Router(config-bdomain)# mac limit maximum addresses 1000
Router(config-bdomain)# end
```

Example Configuring a MAC Address Limit on a Service Instance

```
Router> enable
Router# configure terminal
Router(config)# interface gigabitethernet 3/0/1
Router(config-if)# service instance 100 ethernet
Router(config-if-srv)# encapsulation dot1Q 100
Router(config-if-srv)# bridge-domain 100
Router(config-if-srv)# mac security maximum addresses 10
Router(config-if-srv)# mac security
Router(config-if-srv)# end
```

Example Configuring a MAC Address Violation Response

```
Router> enable
Router# configure terminal
```

```
Router(config)# interface gigabitethernet 3/0/1
Router(config-if)# service instance 100 ethernet
Router(config-if-srv)# encapsulation dot1Q 100
Router(config-if-srv)# bridge-domain 100
Router(config-if-srv)# mac security address permit a2aa.aaaa.aaaa
Router(config-if-srv)# mac security violation protect
Router(config-if-srv)# mac security
Router(config-if-srv)# end
```

Example Configuring MAC Address Aging

```
Router> enable
Router# configure terminal
Router(config)# interface gigabitethernet 4/0/1
Router(config-if)# service instance 100 ethernet
Router(config-if-srv)# encapsulation dot1q 100
Router(config-if-srv)# bridge-domain 100
Router(config-if-srv)# mac security aging time 10
Router(config-if-srv)# mac security
Router(config-if-srv)# end
```

Example Configuring a Sticky MAC Address

```
Router> enable
Router# configure terminal
Router(config)# interface gigabitethernet 3/0/1
Router(config-if)# service instance 100 ethernet
Router(config-if-srv)# encapsulation dot1Q 100
Router(config-if-srv)# bridge-domain 100
Router(config-if-srv)# mac security sticky
Router(config-if-srv)# mac security
```

Example Displaying the MAC Addresses on a Specific Secure Service Instance

```
Router# show ethernet service instance id 1879665131 interface gigabitethernet 0/2 mac
security address
MAC Address       Type      Rem. Age(min)
0001.0001.0001    static    100
0001.0001.0002    static    100
0001.0001.aaaa    dynamic   100
0001.0001.aaab    dynamic   100
```

The table below describes the significant fields in the output.

Table 11 MAC Addresses on a Specific Service Instance: Field Descriptions

Field	Description
MAC Address	Displays the MAC addresses on the service instance.
Type	Indicates the type of MAC address by declaring if it was statically configured (static) or dynamically learned (dynamic).

Field	Description
Rem. Age(min)	Displays the remaining age of the address in minutes. A hyphen (-) indicates that the aging is not enabled.

Note For some platforms such as Cisco 7600 series routers, the MAC address remaining age time information is available only on the switch console. Use the **remote command switch**command and the **show ethernet service instance id id interface** *type number* **mac security address** command to display the MAC address remaining age time information.

Example Displaying the Last Violation on a Specific Service Instance

```
Router# show ethernet service instance id 1879665131 interface gigabitethernet 0/2 mac
security last violation
At: Apr 4 06:57:25.971
Source address: ae4e.b7b5.79ae
Reason: Denied address
```

Example Displaying the MAC Security Status of a Specific Service Instance

```
Router# show ethernet service instance id 1879665131 interface Ethernet0/2 mac security
MAC Security: enabled
```

Example Displaying the MAC Addresses of All Secured Service Instances

```
Router# show ethernet service instance mac security address
Port                  Bridge-domain    MAC Address       Type      Rem. Age(min)
Gi1/0/0 ServInst 1    10               0001.0001.0001    static    82
Gi1/0/0 ServInst 1    10               0001.0001.0002    static    82
Gi1/0/0 ServInst 1    10               0001.0001.aaaa    dynamic   82
Gi1/0/0 ServInst 1    10               0001.0001.aaab    dynamic   82
Gi1/0/0 ServInst 2    10               0002.0002.0002    static    -
Gi1/0/0 ServInst 2    10               0002.0002.0003    static    -
Gi1/0/0 ServInst 2    10               0002.0002.0004    static    -
Gi1/0/0 ServInst 2    10               0002.0002.aaaa    dynamic   -
Gi1/0/0 ServInst 2    10               0002.0002.bbbb    dynamic   -
Gi1/0/0 ServInst 2    10               0002.0002.cccc    dynamic   -
Gi3/0/5 ServInst 10   30               0003.0003.0001    static    200
Gi3/0/5 ServInst 10   30               0003.0003.0002    static    200
```

The table below describes the significant fields in the output.

Table 12 MAC Addresses of All Service Instances: Field Descriptions

Field	Description
Port	Displays the service instance ID number and its interface type and number.
Bridge- Domain	Displays the bridge- domain ID number for each service instance listed.

Field	Description
MAC Address	Displays the MAC addresses on the service instance.
Type	Indicates the type of MAC address by declaring if it was statically configured (static) or dynamically learned (dynamic).
Rem. Age(min)	Displays the remaining age of the address, in minutes. A hyphen (-) indicates that the aging is not enabled.

Note For some platforms such as Cisco 7600 series routers, the MAC address remaining age time information is available only on the switch console. Use the **remote command switch**command and the **show ethernet service instance mac security address**command to display the MAC address remaining age time information.

Example Displaying the MAC Security Statistics of All Service Instances

In the following example, the numbers of allowed and actual secured addresses recorded on the service instance are displayed.

```
Router# show ethernet service instance mac security statistics
Ethernet0/0 service instance 890597333 (bridge-domain 730)
Secure addresses: 3
Address limit: 7
Ethernet0/0 service instance 1559665780 (bridge-domain 1249)
Secure addresses: 8
Address limit: 8
Ethernet0/0 service instance 1877043343 (bridge-domain 1155)
Secure addresses: 0
Address limit: 8
Ethernet0/1 service instance 127771402 (bridge-domain 730)
Secure addresses: 12
Address limit: 12
Ethernet0/1 service instance 183598286 (bridge-domain 730)
Secure addresses: 1
Address limit: 1
Ethernet0/1 service instance 433365207 (bridge-domain 1249)
Secure addresses: 0
Address limit: 1
Ethernet0/1 service instance 858688453 (bridge-domain 1328)
Secure addresses: 0
Address limit: 2
```

Example Displaying the MAC Addresses on All Service Instances for a Bridge Domain

```
Router# show bridge-domain 730 mac security address
Port                 MAC Address       Type      Rem. Age(min)
Gi1/0/0 ServInst 1    0001.0001.0001    static    74
Gi1/0/0 ServInst 1    0001.0001.0002    static    74
Gi1/0/0 ServInst 1    0001.0001.aaaa    dynamic   74
Gi1/0/0 ServInst 1    0001.0001.aaab    dynamic   74
```

```
Gi1/0/0 ServInst 2    0002.0002.0002    static    -
Gi1/0/0 ServInst 2    0002.0002.0003    static    -
Gi1/0/0 ServInst 2    0002.0002.0004    static    -
Gi1/0/0 ServInst 2    0002.0002.aaaa    dynamic   -
Gi1/0/0 ServInst 2    0002.0002.bbbb    dynamic   -
Gi1/0/0 ServInst 2    0002.0002.cccc    dynamic   -
```

Note For some platforms such as Cisco 7600 routers, the MAC address remaining age time information is available only on the switch console. Use the **remote command switch**command and the **show bridge-domain** *id* **mac security address** command to display the MAC address remaining age time information.

Example Displaying the Secured Service Instances for a Specific Bridge Domain

```
Router# show bridge-domain 730 mac security
Gi1/0/0 ServInst 1
MAC Security enabled: yes
Gi1/0/0 ServInst 2
MAC Security enabled: yes
```

Additional References

Related Documents

Related Topic	Document Title
Carrier Ethernet configuration guide	*Cisco IOS Carrier Ethernet Configuration Guide*, Release 12.2SR
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
MEF 6.1	Metro Ethernet Services Definitions Phase 2 (PDF 6/08)
MEF 10.1	Ethernet Services Attributes Phase 2 (PDF 10/06)

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	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, 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 MAC Address Limiting on Service Instances Bridge Domains and EVC Port Channels

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 13 Feature Information for MAC Address Limiting on Service Instances, Bridge Domains, and EVC Port Channels

Feature Name	Releases	Feature Information
MAC Address Limiting on Service Instances and Bridge Domains	12.2(33)SRD	The MAC Address Limiting on Service Instances and Bridge Domains feature addresses port security with service instances by providing the capability to control and filter MAC address learning behavior at the granularity of a per-service instance. When a violation requires a shutdown, only the customer that is assigned to a given service instance is affected. MAC address limiting is a type of MAC security and is also referred to as a MAC security component or element. The following commands were introduced or modified: **bridge-domain (config), bridge-domain (service instance), clear bridge-domain mac table, clear ethernet service instance, errdisable recovery cause mac-security, interface, mac limit maximum addresses, mac security, show bridge-domain, show ethernet service instance**.
MAC Address Security on EVC Port Channel	12.2(33)SRE	The MAC Address Security on EVC Port Channel feature supports MPBE, local connect, and xconnect service types. Load balancing is done on an EFP basis where a number of EFPs exclusively pass traffic through member links. The following command was introduced or modified: **interface**

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Static MAC Address Support on Service Instances and Pseudowires

The Static MAC Address Support on Service Instances and Pseudowires feature supports configuration of a static MAC address on a pseudoport. Use of a static MAC address for broadband network gateway (BNG) upstream traffic enables traffic forwarding while conserving MAC table resources and limiting the traffic flood by creating multicast groups.

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 Static MAC Address Support on Service Instances and Pseudowires

- Knowledge of both port and bridge domain limitations.
- Knowledge of service instances.
- Layer 2 virtual forwarding instance (L2VFI) must be integrated with the bridge domain.

Restrictions for Static MAC Address Support on Service Instances and Pseudowires

- Multicast static MAC addresses are not allowed in MAC address security configurations.
- Static MAC addresses are programmed only on switch processors (both active and standby).

Information About Static MAC Address Support on Service Instances and Pseudowires

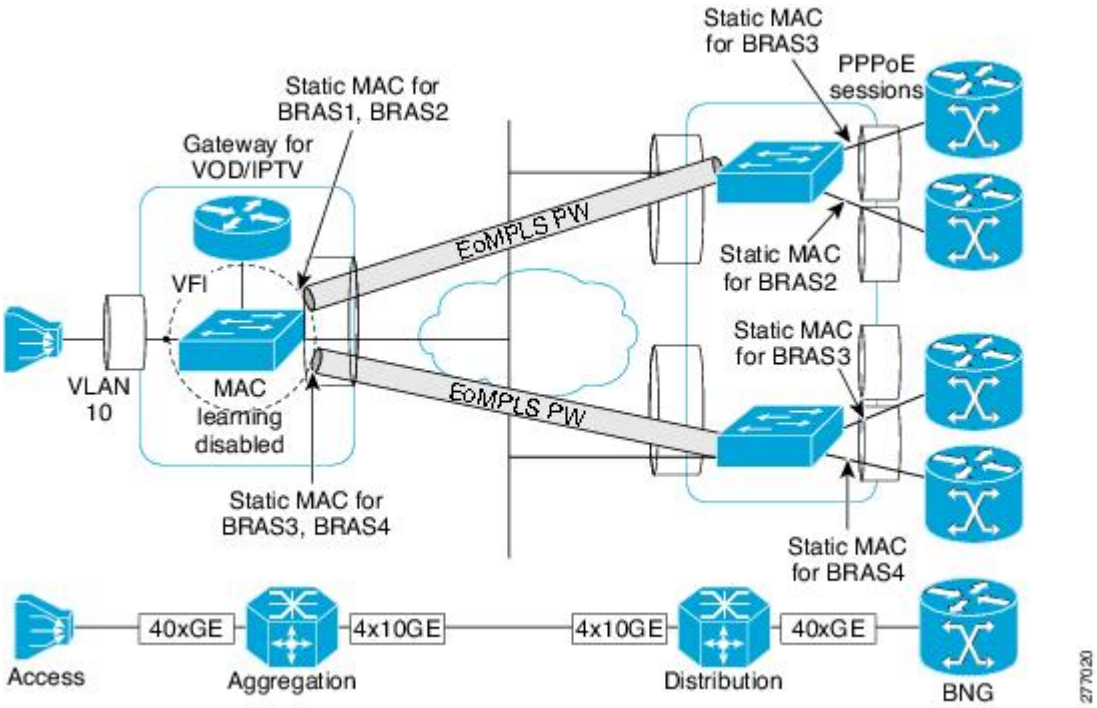
Static MAC Address Support on Service Instances and Pseudowires

Static MAC address configuration on service instances and pseudowires eliminates the need for MAC address learning, which is required for traffic forwarding. In the upstream direction, without MAC address learning, MAC address table resources can be conserved and network resources optimized.

Static MAC address configuration requires L2VFI integration with a bridge domain, which allows a pseudoport to be created on the bridge domain for a pseudowire. After the pseudoport is created, the static MAC configuration can be associated to the bridge domain pseudoport.

Multicast static MAC addresses are allowed on multiple pseudoports in the same bridge domain.

The figure below shows static MAC addresses in a network configured with broadband remote access server (BRAS) redundancy.



When a bridge domain ID is either changed or deleted for a service instance or for an L2VFI, all static MAC addresses are removed.

When a service instance or a pseudowire is deleted, all static MAC addresses on that pseudoport are removed.

Benefits of Static MAC Address Support on Service Instances and Pseudowires

- Facilitates optimization of network resources
- Conserves MAC table resources when used for upstream traffic

How to Configure a Static MAC Address on Service Instances or Pseudowires

Configuring a Static MAC Address on a Service Instance

Perform this task to manually configure a static MAC address on a service instance.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance** *id* **ethernet** [*evc-id*]
5. **encapsulation dot1q** *vlan-id* [**,** *vlan-id*[**-** *vlan-id*]] [**native**]
6. **bridge-domain** *bridge-id* [**split-horizon**[**group** *group-id*]]
7. **mac static address** *mac-addr* [**auto-learn**] [**disable-snooping**]
8. **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 ethernet 1/0`	Configures an interface type and enters interface configuration mode.
Step 4	**service instance** *id* **ethernet** [*evc-id*] **Example:** `Router(config-if)# service instance 1 ethernet`	Configures an Ethernet service instance on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* [**,** *vlan-id*[**-** *vlan-id*]] [**native**] **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
Step 6	**bridge-domain** *bridge-id* [**split-horizon**[**group** *group-id*]] **Example:** `Router(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 7	**mac static address** *mac-addr* [**auto-learn**] [**disable-snooping**] **Example:** `Router(config-if-srv)# mac static address 0000.bbbb.cccc`	Configures a static MAC address.
Step 8	**exit** **Example:** `Router(config-if-srv)# exit`	Returns the CLI to privileged EXEC mode.

Configuring a Static MAC Address on a Pseudowire

Perform this task to manually configure a static MAC address on a Pseudowires.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *name* **manual**
4. **vpn** {**vrf** *vrf-name*| **id** *vpn-id*}
5. **bridge-domain** *bridge-id* **vlan** *vlan-name*
6. **neighbor** *remote-router-id vc-id* {**encapsulation** *encapsulation-type* | **pw-class** *pw-name*} [**no-split-horizon**]
7. **mac static address** *mac-addr* [**auto-learn**] [**disable-snooping**]
8. **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	**l2 vfi** *name* **manual** **Example:** `Router(config)# l2 vfi test-core manual`	Creates a Layer 2 VFI and enters Layer 2 VFI manual configuration mode.
Step 4	**vpn** {**vrf** *vrf-name*| **id** *vpn-id*} **Example:** `Router(config-vfi)# vpn id 100`	Specifies that the source and destination IP addresses of a virtual private dialup network (VPDN) group belong to a specified Virtual Private Network (VPN) routing and forwarding (VRF) instance,
Step 5	**bridge-domain** *bridge-id* **vlan** *vlan-name* **Example:** `Router(config-vfi)# bridge-domain 100 vlan vlan10`	Configures a VLAN for a bridge domain.

	Command or Action	Purpose
Step 6	**neighbor** *remote-router-id vc-id* {**encapsulation** *encapsulation-type* | **pw-class** *pw-name*} [**no-split-horizon**] **Example:** `Router(config-vfi)# neighbor 209.165.202.129 5 pw-class TestClass`	Specifies the type of tunnel signaling and encapsulation mechanism for each virtual private LAN service (VPLS) peer and enters VFI neighbor configuration mode.
Step 7	**mac static address** *mac-addr* [**auto-learn**] [**disable-snooping**] **Example:** `Router(config-vfi-neighbor)# mac static address 0000.aaaa.bbbb`	Configures a static MAC address.
Step 8	**exit** **Example:** `Router(config-vfi-neighbor)# exit`	Returns the CLI to privileged EXEC mode.

Displaying Configured Static MAC Addresses

Perform this task to display the static MAC addresses that are configured. Output of these commands may be useful for troubleshooting. The **show** commands can be issued in any order.

SUMMARY STEPS

1. **enable**
2. **show bridge-domain** [[*bridge-id*] [**c-mac**] [**mac**{**security** [**address** | **last violation** | **statistics**] | **static address**| **table**[*mac-address* | **aging-time** | **count**]}] | **split-horizon** [**group** {*group-number* | **all** | **none**}] | **stats**]
3. **show ethernet service instance** [**detail** | **id** *id* **interface** *type number* [**detail** | **mac** {**security** [**address** | **last violation** | **statistics**] | **static address**}] | **platform** | **stats**] | **interface** *type number* [**detail** | **platform** | **stats** | **summary**] | **mac security** [**address** | **last violation** | **statistics**] | **platform** | **policy-map** | **stats** | **summary**]
4. **show vfi** [**checkpoint** [**summary**] | **mac static address** | **memory** [**detail**] | **name** *vfi-name* [**checkpoint** | **mac static address**] | **neighbor** *ip-addr* **vcid** *vcid* **mac static address**]
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	**show bridge-domain** [[*bridge-id*] [**c-mac**] [**mac**{**security** [**address** | **last violation** | **statistics**] | **static address**| **table**[*mac-address* | **aging-time** | **count**]}] | **split-horizon** [**group** {*group-number* | **all** | **none**}] | **stats**] **Example:** `Router# show bridge-domain 100 mac static address`	Display bridge-domain information.
Step 3	**show ethernet service instance** [**detail** | **id** *id* **interface** *type number* [**detail** | **mac** {**security** [**address** | **last violation** | **statistics**] | **static address**}] | **platform** | **stats**] | **interface** *type number* [**detail** | **platform** | **stats** | **summary**] | **mac security** [**address** | **last violation** | **statistics**] | **platform** | **policy-map** | **stats** | **summary**] **Example:** `Router# show ethernet service instance id 1 interface ethernet 0/0 mac static address`	Displays information about Ethernet service instances.
Step 4	**show vfi** [**checkpoint** [**summary**] | **mac static address** | **memory** [**detail**] | **name** *vfi-name* [**checkpoint** | **mac static address**] | **neighbor** *ip-addr* **vcid** *vcid* **mac static address**] **Example:** `Router# show vfi name VFI2 mac static address`	Displays information about a VFI.
Step 5	**exit** **Example:** `Router# exit`	Returns the CLI to user EXEC mode.

Configuration Examples for Static MAC Address Support on Service Instances and Pseudowires

- Example Configuring a Static MAC Address on a Pseudowire, page 368

Example Configuring a Static MAC Address on a Service Instance

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 1/0
Router(config-if)# service instance 1 ethernet
Router(config-if-srv)# encapsulation dot1q 100
Router(config-if-srv)# bridge-domain 100
Router(config-if-srv)# mac static address 0000.bbbb.cccc
Router(config-if-srv)# exit
```

Example Configuring a Static MAC Address on a Pseudowire

```
Router> enable
Router# configure terminal
Router(config)# l2 vfi test-core manual
Router(config-vfi)# vpn id 100
Router(config-vfi)# bridge-domain 100 vlan vlan10
Router(config-vfi)# neighbor 209.165.202.129 5 pw-class TestClass
Router(config-vfi-neighbor)# mac static address 0000.aaaa.bbbb
Router(config-vfi-neighbor)# exit
```

Additional References

Related Documents

Related Topic	Document Title
Configuration guide	*Cisco IOS Carrier Ethernet Configuration Guide*, Release 12.2SR
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

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
None	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
None	--

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 Static MAC Address Support on Service Instances and Pseudowires

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 14 Feature Information for Static MAC Address Support on Service Instances and Pseudowires

Feature Name	Releases	Feature Information
Static Mac for Open (Infrastructure)	12.2(33)SRE	The Static MAC Address Support on Service Instances and Pseudowires feature supports configuration of a static MAC address on a pseudoport. Use of a static MAC address for BNG upstream traffic enables traffic forwarding while conserving MAC table resources and limiting traffic flooding by creating multicast groups. The following commands were introduced or modified: **mac static address**, **neighbor**, **show bridge domain**, **show ethernet service instance**, **show vfi**.

IEEE 802.1ah on Provider Backbone Bridges

The IEEE 802.1ah on Provider Backbone Bridges feature enables MAC-in-MAC tunneling on Ethernet virtual circuits (EVCs).

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 IEEE 802.1ah on Provider Backbone Bridges

- The router configuration must include an ES40 line card, because the Institute of Electrical and Electronic Engineers (IEEE) 802.1ah standard is supported on ES40 line cards only.
- IEEE 802.1ah is supported on EVC architecture only.

Restrictions for IEEE 802.1ah on Provider Backbone Bridges

- The following features are not supported:
  - Connectivity Fault Management (CFM) over 802.1ah
  - Internet Group Multicast Protocol (IGMP) snooping or any mulitcast protocol on the customer-bridge (c-bridge) domain

  - Standalone customer-facing backbone edge bridge (I-BEB)
  - Standalone backbone core bridge-facing backbone edge bridge (B-BEB)
- The following limits apply to this feature:
  - Maximum number of MAC tunnels is 4094.
  - Maximum number of service instances under MAC tunnels is 16,384.
  - Maximum number of Ethernet Flow Points (EFP) is 32,768.
  - Maximum number of EFPs on a single interface is 8000.
  - 802.1ah on the port channel is supported for one member link per port channel only.

Information About IEEE 802.1ah on Provider Backbone Bridges

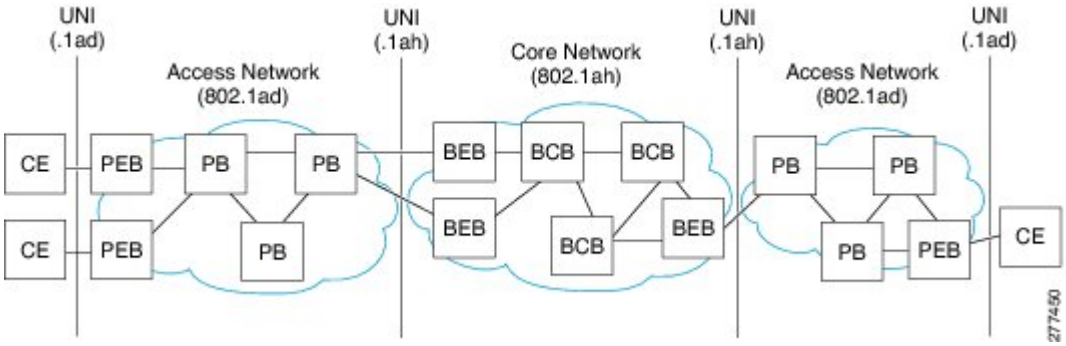
MAC-in-MAC

The IEEE 802.1ah on Provider Backbone Bridges feature encapsulates the end users traffic inside the service providers MAC header, enabling the backbone edge bridge (BEB) to support large numbers of service instances. This functionality is also known as MAC-in-MAC or MAC Tunneling Protocol (MTP). It also allows service providers to hide the identity of their equipment vendors by using user-specified MAC address as the tunnel source address. It also separates the user MAC address space from the provider MAC address space which means that only the edge bridges are aware of the customer MAC addresses, and that only the core bridges are aware of the provider addresses.

The figure below shows a typical 802.1ah PBB network and the table below describes the PBB network components.

Table 15 IEEE 802.1ah PBB Components

Component	Description
BCB	Backbone core bridge
BEB	Backbone edge bridge
CE	Customer equipment
PB	Provider bridge
PEB	Provider edge bridge



Backbone Edge Bridges

BEBs can contain either an I-Component or a B-Component. The I-Component maps Service VLAN identifiers (S-VIDs) to service instance identifiers (I-SIDs) and adds a PBB header without a B-Tag. The B-Component maps I-SIDs to backbone VIDs (B-VIDs) and adds a PBB header with a B-Tag. The IEEE 802.1ah standard specifies the following three types of BEBs:

- The B-Bridge (B-BEB) contains the B-Component of the MAC-in-MAC bridge. It validates the I-SIDs and maps the frames onto the backbone VLAN (B-VLAN). It also switches traffic based on the B-VLANS within the core bridge.
- The I-Bridge (I-BEB) contains the I-Component of the MAC-in-MAC bridge. It performs B-MAC encapsulation and inserts the I-SIDs based on the S-tags, C-tags, or S-tag/C-tag pairs.
- The IB-Bridge (IB-BEB) contains one or more I-Components and a single B-Component interconnected via a LAN segment.

Note The Cisco 7600 series routers are designed to work as IB-Bridges.

IB-Bridges

The IB-Bridge contains both the I-Component and the B-Component. The bridge selects the B-MAC and inserts the I-SID based on the provider VLAN tag (S-tag), the customer VLAN tag (C-tag), or both the S-tag and the C-tag. It validates the I-SIDs and it transmits and receives frames on the B-VLAN.

The IB-Bridge has two types of interfaces:

- Port-based interface: On port-based interfaces all S-tagged frames received from a customer are mapped to an I-SID and the S-tags are preserved.
- S-tagged interface: S-tagged interfaces support one-to-one mapping of an S-VLAN to an I-SID to provide S-VLAN translation capabilities. They also support many-to-one mapping of S-VLANs to an I-SID to provide S-VLAN bundling capability.

The IEEE 802.1ah on Provider Backbone Bridges feature supports all services mandated by the IEEE 802.1ah standard and extends the services to provide additional functionality as follows:

- S-Tagged Service:
  - In multiplexed environments each S-tag maps to an I-SID and may be retained or removed.
  - In bundled environments multiple S-tags map to the same I-SID and the S-tags must be retained.
- C-Tagged Service:

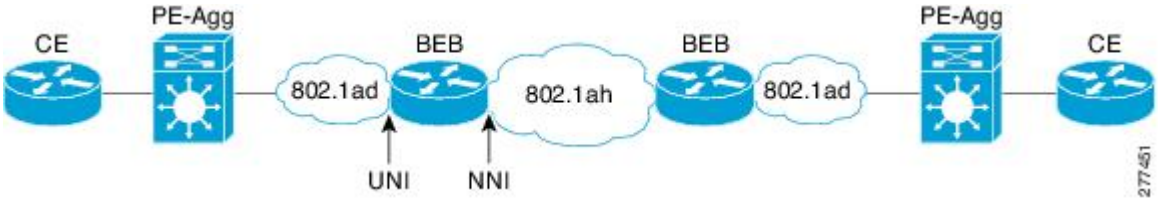
  - In multiplexed environments each C-tag maps to an I-SID and may be retained or removed.
  - In bundled environments multiple C-tags map to the same I-SID and the C-tags must be retained.
- S/C-Tagged Service:
  - In multiplexed environments each S-tag/C-tag pair maps to an I-SID. The S-tag or the S-tag/C-tag pair may be retained or removed.
  - In bundled environments multiple S-tag/C-tags pairs map to the same I-SID and the S-tag/C-tag pair must be retained.
- Port-based Service
  - Any frame whether untagged or double tagged is mapped to the same I-SID and all tags are retained.

IEEE 802.1ah for L2 Bridging Networks

When IEEE 802.1ah is configured on PBBs in an L2 bridging network the packets on the ingress EFP are tunneled to the appropriate MAC tunnel using the bridging identifier in the I-Component (specified using the **bridge-domain c-mac**command). If multiple EFPs use the same I-SID then the C-MAC bridge domain also performs the switching between the EFPs.

The figure below shows a typical L2 bridging network configuration.

Figure 1 IEEE 802.1ah L2 Bridging Network



The table below describes the components of the L2 bridging network.

Table 16 L2 Bridging Network Components

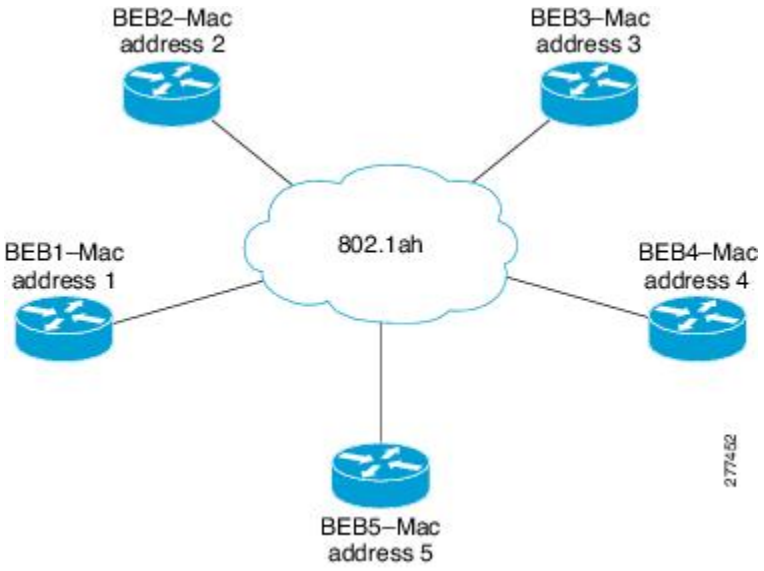
Component Name	Description
802.1ad	IEEE 802.1ad (provider bridges) network
802.1ah	IEEE 802.1ah (provider backbone bridge) network
BEB	Backbone edge bridge
CE	Customer equipment
NNI	Network-to-network interface (egress EFP)
PE-Agg	Provider edge aggregation device
UNI	User-Network Interface (ingress EFP)

- Unknown Unicast and Customer Multicast Traffic, page 375

Unknown Unicast and Customer Multicast Traffic

The figure below shows an L2 network where all the BEBs are connected to each other through a single Backbone VLAN (B-VLAN). In this scenario any unknown unicast traffic from BEB1 is forwarded to BEB2 through to BEB5 because they all share the same B-VLAN.

Figure 2 BEB B-VLAN Network



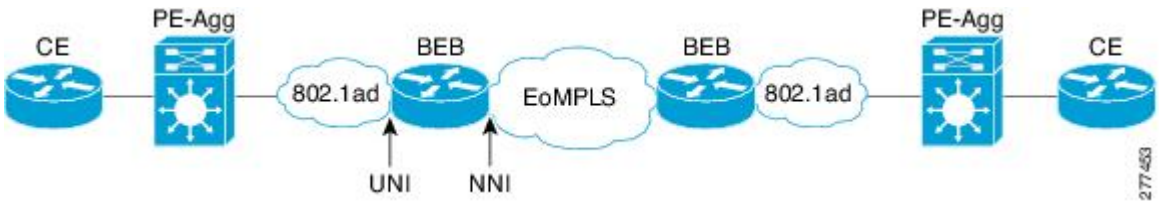
In order to reduce network traffic you can configure a BEB to send traffic to specific BEBs on the B-VLAN. For example, if BEB1 needs to send traffic to BEB3 and BEB4 only, you can use the **mac tunnel address destination map** command to map the customer destination address (C-DA) to a multicast backbone destination address (B-DA). BEB3 and BEB4 are then registered to receive traffic for this B-DA. All packets within the 802.1ah network must be sent to a specified MAC address. The address is a static entry in the MAC address tables in the backbone core bridges. If a default MAC tunnel address is not specified in the table, then all unknown unicast packets and customer multicast traffic are sent with the default B-DA, which is a combination of IEEE-assigned Organizational Unique Identifier (OUI) and the I-SID values.

IEEE 802.1ah for Ethernet Over MPLS

When IEEE 802.1ah is configured on Ethernet over Multiprotocol Label Switching (EoMPLS) networks, the Ethernet links are transported as pseudowires using MPLS label switched paths (LSPs) inside an MPLS tunnel. To configure MAC-in-MAC on EoMPLS networks you must specify ingress EFP configuration settings at the UNI, specify MAC-in-MAC settings, and specify switch virtual interface (SVI) configuration settings at the egress NNI. The SVI represents a VLAN of switch ports connected to the bridge via a single interface.

The figure below shows a typical EoMPLS network configuration.

Figure 3 EEE 802.1ah EoMPLS Network



Note In EoMPLS networks Cisco 7600 series routers use the bridge domain identifier (set using the **bridge-domain** command) as the B-tag identifier. Therefore it is not necessary to specify B-VLAN configuration for the MAC-in-MAC tunnel.

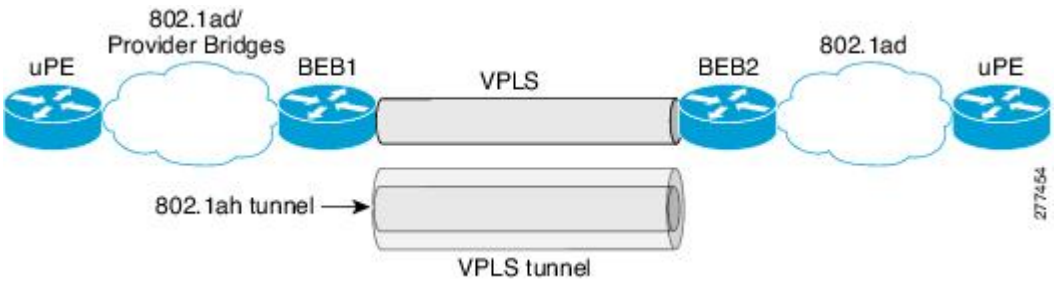
IEEE 802.1ah for Virtual Private LAN Services

When IEEE 802.1ah is configured on virtual private LAN service (VPLS) networks the 802.1ah packets are encapsulated in the VPLS pseudowire.

To configure MAC-in-MAC on VPLS networks you must specify the ingress EFP configuration settings at the UNI, specify the MAC-in-MAC settings, specify the virtual forwarding interface (VFI) settings, and specify the SVI configuration settings at the egress NNI. The SVI represents a VLAN of switch ports connected to the bridge via a single interface.

The figure below shows two 802.1ah networks connected by VPLS.

Figure 4 IEEE 802.1ah VPLS Network



How to Configure MAC-in-MAC on Provider Backbone Bridges

Configuring MAC-in-MAC in an L2 Bridging Network

Perform this task to configure MAC-in-MAC in an L2 bridging network where the NNI has a switchport-based configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *port*
4. **service instance** *id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **bridge-domain** *bridge-id* **c-mac**
7. **exit**
8. **exit**
9. **ethernet mac-tunnel virtual** *tunnel-id*
10. **description** *description*
11. **bridge-domain** *bridge-id*
12. **mac tunnel address destination default** *mac-addr*
13. **service instance** *id* **ethernet**
14. **encapsulation dot1ah isid** *isid*
15. **mac tunnel address destination map** *c-mac-addr b-mac-addr*
16. **bridge-domain** *bridge-id* **c-mac**
17. **exit**
18. **exit**
19. **interface gigabitethernet** *slot* / *port*
20. **switchport**
21. **switchport mode trunk**
22. **switchport trunk allowed vlan** *vlan-id*
23. **end**
24. **show bridge-domain**
25. **show ethernet mac-tunnel engine slot** *slot-number*
26. **show ethernet service instance**
27. **show ethernet service mac-tunnel**

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 gigabitethernet** *slot* / *port* **Example:** `Router(config)# interface gigabitethernet 6/1`	Specifies the Gigabit Ethernet interface to configure as the customer instance port and enters interface configuration mode.
Step 4	**service instance** *id* **ethernet** **Example:** `Router(config-if)# service instance 101 ethernet`	Creates an L2 service instance on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 13`	Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.
Step 6	**bridge-domain** *bridge-id* **c-mac** **Example:** `Router(config-if-srv)# bridge-domain 12 c-mac`	Specifies the bridging identifier in the I-Component.
Step 7	**exit** **Example:** `Router(config-if-srv)# exit`	Exits service instance configuration mode.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits service interface configuration mode.

	Command or Action	Purpose
Step 9	**ethernet mac-tunnel virtual** *tunnel-id* **Example:** `Router(config)# ethernet mac-tunnel virtual 1`	Configures a virtual MAC-in-MAC tunnel and enters MAC-in-MAC tunnel configuration mode.
Step 10	**description** *description* **Example:** `Router(config-tunnel-minm)# description MAC-Tunnel-1`	(Optional) Describes the name and purpose of the MAC tunnel.
Step 11	**bridge-domain** *bridge-id* **Example:** `Router(config-tunnel-minm)# bridge-domain 100`	Binds the MAC tunnel to the bridge domain instance.
Step 12	**mac tunnel address destination default** *mac-addr* **Example:** `Router(config-tunnel-minm)# mac tunnel address destination default 4444.1111.1111`	Specifies a B-DA for a group of service instance IDs (I-SIDs).
Step 13	**service instance** *id* **ethernet** **Example:** `Router(config-tunnel-minm)# service instance 10 ethernet`	Defines an EFP that corresponds to a specific I-SID encapsulation and enters tunnel service configuration mode.
Step 14	**encapsulation dot1ah isid** *isid* **Example:** `Router(config-tunnel-srv)# encapsulation dot1ah isid 10000`	Configures dot1ah encapsulation for the specified I-SID.
Step 15	**mac tunnel address destination map** *c-mac-addr b-mac-addr* **Example:** `Router(config-tunnel-srv)# mac tunnel address destination map 3333.1111.1111 5555.2222.2222`	Maps the service provider backbone bridge MAC address to a customer MAC address.

	Command or Action	Purpose
Step 16	**bridge-domain** *bridge-id* **c-mac** **Example:** `Router(config-tunnel-srv)# bridge-domain 30 c-mac`	Configures the bridge domain as a customer domain.
Step 17	**exit** **Example:** `Router(config-tunnel-srv)# exit`	Exits tunnel service configuration mode.
Step 18	**exit** **Example:** `Router(config-tunnel-minm)# exit`	Exits MAC-in-MAC tunnel configuration mode.
Step 19	**interface gigabitethernet** *slot* / *port* **Example:** `Router(config)# interface gigabitethernet 6/2`	Specifies the Gigabit Ethernet interface to configure as the bridge instance port and enters interface configuration mode.
Step 20	**switchport** **Example:** `Router(config-if)# switchport`	Modifies the switching characteristics of the L2 switched interface.
Step 21	**switchport mode trunk** **Example:** `Router(config-if)# switchport mode trunk`	Specifies a trunking VLAN L2 interface.
Step 22	**switchport trunk allowed vlan** *vlan-id* **Example:** `Router(config-if)# switchport trunk allowed vlan 100`	Sets the list of allowed VLANs that transmit traffic from this interface in tagged format when in trunking mode.

	Command or Action	Purpose
Step 23	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and enables user EXEC mode.
Step 24	**show bridge-domain** **Example:** `Router> show bridge-domain`	(Optional) Displays bridge-domain information.
Step 25	**show ethernet mac-tunnel engine slot** *slot-number* **Example:** `Router> show ethernet mac-tunnel engine slot 2`	(Optional) Displays Ethernet MAC-in-MAC information.
Step 26	**show ethernet service instance** **Example:** `Router> show ethernet service instance`	(Optional) Displays Ethernet service instance information.
Step 27	**show ethernet service mac-tunnel** **Example:** `Router> show ethernet service mac-tunnel`	(Optional) Displays Ethernet service MAC-in-MAC information.

Configuring MAC-in-MAC in an Ethernet over MPLS Network

Perform this task to configure MAC-in-MAC in an EoMPLS network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *port*
4. **service instance** *id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **bridge-domain** *bridge-id* **c-mac**
7. **exit**
8. **exit**
9. **ethernet mac-tunnel virtual** *tunnel-id*
10. **bridge-domain** *bridge-id*
11. **service instance** *id* **ethernet**
12. **encapsulation dot1ah isid** *isid*
13. **bridge-domain** *bridge-id* **c-mac**
14. **exit**
15. **exit**
16. **interface vlan** *vlanid*
17. **xconnect** *ipaddress vc-id* **encapsulation mpls**
18. **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 gigabitethernet** *slot* / *port* **Example:** `Router(config)# interface gigabitethernet 6/1`	Specifies the Gigabit Ethernet interface to configure as the customer instance port and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**service instance** *id* **ethernet** **Example:** `Router(config-if)# service instance 101 ethernet`	Creates an L2 service instance on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 13`	Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.
Step 6	**bridge-domain** *bridge-id* **c-mac** **Example:** `Router(config-if-srv)# bridge-domain 12 c-mac`	Specifies the bridging identifier in the I-Component.
Step 7	**exit** **Example:** `Router(config-if-srv)# exit`	Exits service instance configuration mode.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 9	**ethernet mac-tunnel virtual** *tunnel-id* **Example:** `Router(config)# ethernet mac-tunnel virtual 1`	Configures a virtual MAC-in-MAC tunnel and enters MAC-in-MAC tunnel configuration mode.
Step 10	**bridge-domain** *bridge-id* **Example:** `Router(config-tunnel-minm)# bridge-domain 100`	Binds the MAC tunnel to the bridge domain instance.

	Command or Action	Purpose
Step 11	**service instance** *id* **ethernet** **Example:** `Router(config-tunnel-minm)# service instance 10 ethernet`	Defines an EFP that corresponds to a specific I-SID encapsulation and enters tunnel service configuration mode.
Step 12	**encapsulation dot1ah isid** *isid* **Example:** `Router(config-tunnel-srv)# encapsulation dot1ah isid 10000`	Configures dot1ah encapsulation for the specified I-SID.
Step 13	**bridge-domain** *bridge-id* **c-mac** **Example:** `Router(config-tunnel-srv)# bridge-domain 30 c-mac`	Configures the bridge domain as a customer domain.
Step 14	**exit** **Example:** `Router(config-tunnel-srv)# exit`	Exits tunnel service configuration mode.
Step 15	**exit** **Example:** `Router(config-tunnel-minm)# exit`	Exits MAC-in-MAC tunnel configuration mode.
Step 16	**interface vlan** *vlanid* **Example:** `Router(config)# interface vlan 1000`	Creates a dynamic SVI, and enters interface configuration mode.
Step 17	**xconnect** *ipaddress vc-id* **encapsulation mpls** **Example:** `Router(config-if)# xconnect 10.243.245.11 100 encapsulation mpls`	Binds the attachment circuit to the pseudowire, and configures an Any Transport over MPLS (AToM) static pseudowire. • Specifies MPLS as the tunneling method to encapsulate the data in the pseudowire.

	Command or Action	Purpose
Step 18	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.

Configuring MAC-in-MAC in a VPLS Network

Perform this task to configure MAC-in-MAC in a VPLS network. The following configuration enables the router to work as an IB-Bridge.

Note On Cisco 7600 series routers the bridge-domain identifier must be the same as the SVI identifier.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface gigabitethernet** *slot* / *port*
4. **service instance** *id* **ethernet**
5. **encapsulation dot1q** *vlan-id*
6. **bridge-domain** *bridge-id* **c-mac**
7. **exit**
8. **exit**
9. **ethernet mac-tunnel virtual** *tunnel-id*
10. **bridge-domain** *bridge-id*
11. **service instance** *id* **ethernet**
12. **encapsulation dot1ah isid** *isid*
13. **bridge-domain** *bridge-id* **c-mac**
14. **exit**
15. **service instance** *id* **ethernet**
16. **encapsulation dot1ah isid** *isid*
17. **bridge-domain** *bridge-id* **c-mac**
18. **exit**
19. **exit**
20. **l2 vfi** *vfi-name* **manual**
21. **vpn id** *vpn-id*
22. **neighbor** *ipaddress vcid* **encapsulation mpls**
23. **neighbor** *ipaddress vcid* **encapsulation mpls**
24. **exit**
25. **interface vlan** *vlanid*
26. **xconnect** *ipaddress vc-id* **encapsulation mpls**
27. **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 gigabitethernet** *slot* / *port* **Example:** `Router(config)# interface gigabitethernet 6/1`	Specifies the Gigabit Ethernet interface to configure as the customer instance port and enters interface configuration mode.
Step 4	**service instance** *id* **ethernet** **Example:** `Router(config-if)# service instance 101 ethernet`	Creates an L2 service instance on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* **Example:** `Router(config-if-srv)# encapsulation dot1q 13`	Defines the matching criteria to be used in order to map ingress dot1q frames on an interface to the appropriate service instance.
Step 6	**bridge-domain** *bridge-id* **c-mac** **Example:** `Router(config-if-srv)# bridge-domain 12`	Specifies the bridging identifier in the I-Component.
Step 7	**exit** **Example:** `Router(config-if-srv)# exit`	Exits service instance configuration mode.
Step 8	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.

	Command or Action	Purpose
Step 9	**ethernet mac-tunnel virtual** *tunnel-id* **Example:** `Router(config)# ethernet mac-tunnel virtual 1`	Configures a virtual MAC-in-MAC tunnel and enters MAC-in-MAC tunnel configuration mode.
Step 10	**bridge-domain** *bridge-id* **Example:** `Router(config-tunnel-minm)# bridge-domain 100`	Binds the MAC tunnel to the bridge domain instance.
Step 11	**service instance** *id* **ethernet** **Example:** `Router(config-tunnel-minm)# service instance 31 ethernet`	Defines an EFP that corresponds to a specific I-SID encapsulation and enters tunnel service configuration mode.
Step 12	**encapsulation dot1ah isid** *isid* **Example:** `Router(config-tunnel-srv)# encapsulation dot1ah isid 10000`	Configures dot1ah encapsulation for the specified I-SID.
Step 13	**bridge-domain** *bridge-id* **c-mac** **Example:** `Router(config-tunnel-srv)# bridge-domain 10 c-mac`	Configures the bridge domain as a customer domain.
Step 14	**exit** **Example:** `Router(config-tunnel-srv)# exit`	Exits tunnel service configuration mode.
Step 15	**service instance** *id* **ethernet** **Example:** `Router(config-tunnel-minm)# service instance 41 ethernet`	Defines an EFP that corresponds to a specific I-SID encapsulation and enters tunnel service configuration mode.

	Command or Action	Purpose
Step 16	**encapsulation dot1ah isid** *isid* **Example:** `Router(config-tunnel-srv)# encapsulation dot1ah isid 20000`	Configures dot1ah encapsulation for the specified I-SID.
Step 17	**bridge-domain** *bridge-id* **c-mac** **Example:** `Router(config-tunnel-srv)# bridge-domain 20 c-mac`	Configures the bridge domain as a customer domain.
Step 18	**exit** **Example:** `Router(config-tunnel-srv)# exit`	Exits tunnel service configuration mode.
Step 19	**exit** **Example:** `Router(config-tunnel-minm)# exit`	Exits MAC-in-MAC tunnel configuration mode.
Step 20	**l2 vfi** *vfi-name* **manual** **Example:** `Router(config)# l2 vfi myvfi manual`	Configures a virtual forwarding instance and enters L2 VFI point-to-point configuration mode.
Step 21	**vpn id** *vpn-id* **Example:** `Router(config-vfi)# vpn id 20`	Sets a VPN ID on a VPN routing and forwarding (VRF) instance.
Step 22	**neighbor** *ipaddress vcid* **encapsulation mpls** **Example:** `Router(config-vfi)# neighbor 172.16.10.12 2000 encapsulation mpls`	Specifies the first router that forms a point-to-point Layer 2 VFI connection.

	Command or Action	Purpose
Step 23	**neighbor** *ipaddress vcid* **encapsulation mpls** **Example:** `Router(config-vfi)# neighbor 172.16.200.120 2000 encapsulation mpls`	Specifies the second router that forms a point-to-point Layer 2 VFI connection.
Step 24	**exit** **Example:** `Router(config-vfi)# exit`	Exits L2 VFI point-to-point configuration mode.
Step 25	**interface vlan** *vlanid* **Example:** `Router(config)# interface vlan 1000`	Creates a dynamic SVI, and enters interface configuration mode.
Step 26	**xconnect** *ipaddress vc-id* **encapsulation mpls** **Example:** `Router(config-if)# xconnect 10.243.245.11 100 encapsulation mpls`	Binds the attachment circuit to the pseudowire, and configures an AToM static pseudowire. • Specifies MPLS as the tunneling method to encapsulate the data in the pseudowire.
Step 27	**exit** **Example:** `Router(config-if)# exit`	Returns to global configuration mode.

Configuration Examples for MAC-in-MAC on Provider Backbone Bridges

Example MAC-in-MAC Configuration for L2 Bridging Networks

In the following example, the UNI configuration is performed on the GigabitEthernet 1/0, GigabitEthernet 2/0, and GigabitEthernet 3/0 interfaces. The MAC-in-MAC tunnel configuration includes commands to

configure the default MAC tunnel destination address and the destination map. The NNI configuration is performed on the GigabitEthernet 1/2 interface, and shows the options for a switchport or External Interface (EI)-based NNI.

Note For switchport NNI configurations the VLAN ID is the same as the bridge domain ID configured under the MAC tunnel. For EI NNI configurations a service instance is configured under the NNI interface and the binding of the MAC tunnel to the service instance is done using the bridge domain.

UNI (Ingress) Configuration

```
interface gigabitethernet  1/0
 service instance 10 ethernet
  encapsulation dot1q 10
  bridge-domain 20 c-mac
 service instance 20 ethernet
  encapsulation dot1q  20
  bridge-domain 30 c-mac
interface gigabitethernet 2/0
 service instance 10 ethernet
  encapsulation dot1q 10
  bridge-domain 20 c-mac
 service instance 30 ethernet
  encapsulation dot1q 20
  bridge-domain 30 c-mac
interface gigabitethernet 3/0
 service instance 10 ethernet
  encapsulation dot1q 10
  bridge-domain 20 c-mac
```

MAC-in-MAC Tunnel Configuration

```
ethernet mac-tunnel virtual 1
 bridge-domain 100
 mac tunnel address destination default 4444.1111.1111
 service instance 10 ethernet
  encapsulation  dot1ah isid 10000
  bridge-domain 20 c-mac
 service instance 20 ethernet
  encapsulation  dot1ah isid 20000
  bridge-domain 30 c-mac
  mac tunnel address destination map 3333.1111.1111 5555.2222.2222
```

Switchport NNI (Egress) Configuration

```
interface gigabitethernet 1/2
 switchport
 switchport mode trunk
 switchport trunk allowed vlan 100
```

EI NNI (Egress) Configuration

```
interface gigabitethernet 1/2
 service instance 20 ethernet
  encapsulation dot1q
  bridge-domain 100
```

Example MAC-in-MAC Configuration for Ethernet over MPLS Networks

The following example shows how to configure a BEB where two 802.1ah networks are connected using MPLS:

UNI (Ingress) Configuration

```
interface gigabitethernet 1/1
 service instance 15 ethernet
  encapsulation dot1q 20
  bridge-domain 10 c-mac
```

MAC-in-MAC Tunnel Configuration

```
ethernet mac-tunnel virtual 1
 bridge-domain 1000
 service instance 500 ethernet
  encapsulation dot1ah isid 10000
  bridge-domain 10 c-mac
```

SVI Configuration

```
interface vlan 1000
 xconnect 10.243.245.11 100 encapsulation mpls
```

Example MAC-in-MAC Configuration for VPLS Networks

The following example shows how to configure a BEB where two 802.1ah networks are connected using VPLS. The 802.1ah packets are encapsulated in the VPLS pseudowire.

UNI (Ingress) Configuration

```
interface gigabitethernet 1/1
 service instance 21 ethernet
  encapsulation dot1q 20
  bridge-domain 10 c-mac
```

MAC-in-MAC Tunnel Configuration

```
ethernet mac-tunnel virtual 1
 bridge-domain 100
 service instance 31 ethernet
  encapsulation dot1ah isid 10000
  bridge-domain 10 c-mac
 service instance 41 ethernet
  encapsulation dot1ah isid 30000
  bridge-domain 20 c-mac
```

VFI Configuration

```
l2 vfi myvfi manual
vpn id 20
 neighbor 172.16.10.12 2000 encapsulation mpls
 neighbor 172.16.200.120 2000 encapsulation mpls
vpn id vpn-id
```

SVI Configuration

```
interface vlan 100
 xconnect vfi vfi100
```

Additional References

Related Documents

Related Topic	Document Title
MAC-in-MAC 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 802.1ah	IEEE 802.1ah - Provider Backbone Bridges

MIBs

MIB	MIBs Link
None	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 IEEE 802.1ah on Provider Backbone Bridges

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 17 Feature Information for IEEE 802.1ah on Provider Backbone Bridges feature.

Feature Name	Releases	Feature Information
802.1ah/EVC2.0 for 7600 (Infrastructure)	12.2(33)SRE	The IEEE 802.1ah on Provider Backbone Bridges feature enables MAC-in-MAC on EVCs. In Cisco IOS Release 12.2(33)SRE, this feature was introduced on the Cisco 7600 series routers. The following commands were introduced or modified: **bridge-domain**, **clear bridge-domain mac table**, **description**, **encapsulation dot1ah isid**, **ethernet mac-tunnel virtual**, **mac tunnel address destination default**, **mac tunnel address destination map**, **service instance ethernet**(mac-tunnel), **show bridge-domain**, **show ethernet mac-tunnel engine slot**, **show ethernet service instance**, **show ethernet service mac-tunnel**.

Enabling Ethernet Local Management Interface

Ethernet Local Management Interface (LMI) is an Ethernet layer operation, administration, and management (OAM) protocol. It provides information that enables autoconfiguration of customer edge (CE) devices and provides the status of Ethernet virtual connections (EVCs) for large Ethernet metropolitan-area networks (MANs) and WANs. Specifically, Ethernet LMI 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 and a user-network interface (UNI) to a CE device.

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 Enabling Ethernet Local Management Interface

Business Requirements

- Ethernet OAM such as connectivity fault management (CFM) must be implemented and operational on the service provider's network.

Restrictions for Enabling Ethernet Local Management Interface

- Ethernet LMI relies on Ethernet CFM for the status of an EVC, the remote UNI identifier associated with an EVC, and remote UNI status.
- Ethernet LMI CE is available only on routing ports on routing platforms. For information about Ethernet LMI PE functionality on switching platforms, see the "Configuring Ethernet CFM and E-LMI" chapter of the *Cisco ME 3400 Switch Software Configuration Guide* , Release 12.2(25)SEG.
- Ethernet LMI in the Cisco IOS Software Release 12.4(9)T does not support autoconfiguration of CE devices.

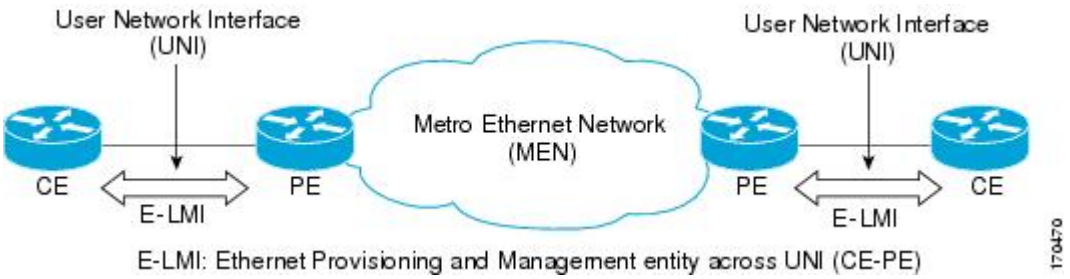
Information About Enabling Ethernet Local Management Interface

EVC

An EVC as defined by the Metro Ethernet Forum could be a port level point-to-point or multipoint-to-multipoint Layer 2 circuit. EVC status can be used by the 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 layer OAM protocol between a CE device and the PE in large Ethernet MANs and WANs. It provides information that enables service providers to autoconfigure CE devices with service parameters and parameter changes from a user provider edge (UPE) device.
The figure below shows where in a network Ethernet LMI functions.



LMI also provides the status of Ethernet EVCs in large Ethernet MANs and WANs to the CE. Specifically, Ethernet LMI 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 EVC and UNI attributes to a CE device.

The Ethernet LMI protocol includes the following procedures, as defined by the MEF 16 Technical Specification:

- Notifying the CE when an EVC is added
- Notifying the CE when an EVC is deleted
- Notifying the CE of the availability state of a configured EVC (Active, Not Active, or Partially Active)
- Communicating UNI and EVC attributes to the CE

- Benefits of Ethernet LMI, page 399

Benefits of Ethernet LMI

- 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

How to Enable Ethernet Local Management Interface

- Enabling Ethernet LMI on All Supported Interfaces, page 399
- Enabling Ethernet LMI on a Single Supported Interface, page 400

Enabling Ethernet LMI on All Supported Interfaces

Perform this task to enable Ethernet LMI on all supported interfaces on a device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet lmi global**
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`	Puts the CLI in global configuration mode.
Step 3	**ethernet lmi global** **Example:** `Router(config)# ethernet lmi global`	Enables Ethernet LMI on all supported interfaces on the device.
Step 4	**end** **Example:** `Router# end`	Returns the CLI to privileged EXEC mode.

Enabling Ethernet LMI on a Single Supported Interface

Perform the steps in this task to enable Ethernet LMI on a specific supported interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ethernet lmi interface**
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`	Puts the CLI in global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface ethernet 0/0`	Specifies an interface and puts the CLI in interface configuration mode.
Step 4	**ethernet lmi interface** **Example:** `Router(config-if)# ethernet lmi interface`	Enables Ethernet LMI on the interface.
Step 5	**end** **Example:** `Router# end`	Returns the CLI to privileged EXEC mode.

Configuration Examples for Ethernet Local Management Interface

The examples in this section show the configurations that enable Ethernet LMI on all interfaces on a CE device (globally) and on a specific interface on a CE device.

- Example Enabling Ethernet LMI on All Supported Interfaces, page 401
- Example Enabling Ethernet LMI on a Single Supported Interface, page 401

Example Enabling Ethernet LMI on All Supported Interfaces

```
enable
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ethernet lmi global
end
00:06:33: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed p
```

Example Enabling Ethernet LMI on a Single Supported Interface

```
enable
configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
interface ethernet 0/0
ethernet lmi interface
end
00:05:51: %SYS-5-CONFIG_I: Configured from console by console
```

Additional References

Related Documents

Related Topic	Document Title
Ethernet Connectivity Fault Management	"Configuring Ethernet Connectivity Fault Management in a Service Provider Network" in the *Cisco IOS Carrier Ethernet Configuration Guide*
Configuring CFM and E-LMI in a service provider network	*Cisco ME 3400 Switch Software Configuration Guide, Rel. 12.2(25)SEG*
Commands used for configuring Ethernet LMI in a service provider network	*Cisco ME 3400 Switch Command Reference, Rel. 12.2(25)SEG*
Ethernet LMI at a provider edge	"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 Command List, All Releases

Standards

Standard	Title
Metro Ethernet Forum 16 Technical Specification	Technical Specification MEF 16- Ethernet Local Management Interface
IEEE P802.1ag/D5.2	*Draft Standard for Local and Metropolitan Area Networks*
ITU-T Q.3/13	Liaison statement on Ethernet OAM (Y.17ethoam)
IETF VPLS OAM	*L2VPN OAM Requirements and Framework*

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 Enabling Ethernet Local Management Interface

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 18 Feature Information for Enabling Ethernet Local Management Interface

Feature Name	Releases	Feature Information
Ethernet Local Management Interface	12.4(9)T 12.2(33)SRB 12.4(15)T2	Ethernet LMI is an Ethernet layer OAM protocol. It provides information that enables autoconfiguration of CE devices and provides the status of EVCs for large Ethernet MANs and WANs. This feature was implemented on the Cisco 7600 router in Cisco IOS Release 12.2(33)SRB. The following commands were introduced or modified: **clear ethernet lmi statistics**, **debug ethernet lmi**, **ethernet lmi**, **ethernet lmi global**, **ethernet lmi interface**, **show ethernet lmi**.

Glossary

CE --customer edge. Edge equipment on the customer side of a user-network interface (UNI).

CE-VLAN ID --Identifier of a CE-VLAN.

E-LMI --Ethernet Local Management Interface. An Ethernet layer OAM protocol. It provides information that enables autoconfiguration of CE devices and provides the status of Ethernet virtual connections (EVCs) for large Ethernet MANs and WANs.

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

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.

PE --provider edge. Edge equipment on the service provider side of a user-network interface (UNI).

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/D5.2 standard when the purpose for various features of LMI are explained.

and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Configuring Remote Port Shutdown

The Remote Port Shutdown feature uses Ethernet Local Management Interface (LMI) in an Ethernet over Multiprotocol Label Switching (EoMPLS) network to propagate remote link status to a customer edge (CE) 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 Remote Port Shutdown

- Ethernet LMI must be enabled for the Remote Port Shutdown feature to function.

Restrictions for Configuring Remote Port Shutdown

- Connectivity Fault Management and Lightweight Directory Protocol (LDP) cannot be configured at the same time.

Information About Configuring Remote Port Shutdown

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 into the service provider network or in some cases, fall back to a backup path over Ethernet or over another alternative service such as Frame Relay or ATM.

Ethernet LMI

Ethernet LMI is an Ethernet Operations, Administration, and Maintenance (OAM) protocol between a CE device and a Provider Edge (PE) device. Ethernet LMI provides information that enables autoconfiguration of CE devices and provides the status of EVCs for large Ethernet metropolitan area networks (MANs) and WANs. Specifically, Ethernet LMI runs only on the PE-CE user network interface (UNI) link and notifies a CE device of both 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 Connectivity Fault Management (CFM) and LDP. In this case Ethernet LMI relies on the OAM manager to interwork with LDP to report remote link status to the local CE.

OAM Manager

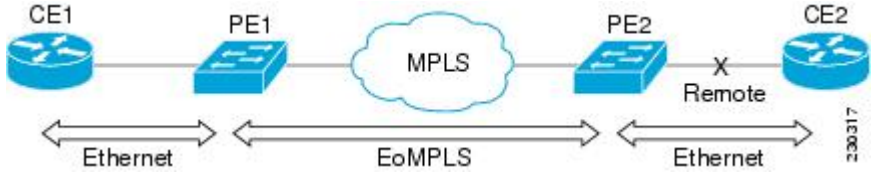
The OAM manager is an infrastructure element that streamlines interaction between OAM protocols. The OAM manager requires two interworking OAM protocols, in this case Ethernet LMI and MPLS LDP.

No interactions are required between Ethernet LMI and the OAM manager on the CE side. On the user-facing provider edge (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 the OAM manager receives notification from the OAM protocol that the EVC status has changed. In this case, the change is called a remote link status change.

Benefits of Remote Port Shutdown

The Remote Port Shutdown feature provides direct interaction of Ethernet LMI with MPLS, LDP, and OAM. When CFM/802.1ag is not running in a network, Remote Port Shutdown enables communication of link status to a CE, and traffic from the CE can be stopped if MPLS or the pseudowire is down. The figure below shows an EoMPLS network with the remote link down.



How to Configure Remote Port Shutdown

Specifying LDP as an OAM Protocol

Perform this task to specify LDP as an OAM protocol.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet evc** *evc-id*
4. **oam protocol** {**cfm svlan** *svlan-id* **domain** *domain-name*| **ldp**}
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	**ethernet evc** *evc-id* **Example:** `Router(config)# ethernet evc evc10`	Defines an EVC and enters EVC configuration mode.
Step 4	**oam protocol** {**cfm svlan** *svlan-id* **domain** *domain-name*| **ldp**} **Example:** `Router(config-evc)# oam protocol ldp`	Configures either CFM or LDP as an OAM protocol. • In this example, LDP is the protocol being configured.

	Command or Action	Purpose
Step 5	**end** **Example:** `Router(config-evc)# end`	Returns the CLI to privileged EXEC mode.

Configuration Examples for Remote Port Shutdown

Example Specifying LDP As the OAM Protocol and Associating a Service Instance to an EVC

In this example, the OAM protocol for EVC pw_evc is specified as LDP, and service instance 1 is associated with the EVC.

```
Router(config)# ethernet evc pw_evc
Router(config-evc)# oam protocol ldp

Router(config-evc)# uni count 2
Router(config-evc)# exit
Router(config)# pseudowire-class vlan-xconnect
Router(config-pw-class)# encapsulation mpls
Router(config-pw-class)# interworking
Router(config-pw-class)# exit
Router(config)# interface ethernet 0/0
Router(config-if)# ethernet lmi interface
Router(config-if)# ethernet uni id ce1
Router(config-if)# service instance 1 ethernet pw_evc
Router(config-if-srv)# encapsulation dot1q 2
Router(config-if-srv)# xconnect10.2.2.2 123 pw-class vlan-xconnect
Router(config_if-srv)# exit
```

Example Configuring Xconnect Directly on an Interface

In this example, Xconnect is configured directly on an interface.

```
Router(config)# interface ethernet 0/0
Router(config-if)# xconnect 2.2.2.2 123 pw-class vlan-xconnect
Router(config-if)# ethernet lmi interface
Router(config-if)# ethernet uni id ce1
Router(config-if)# service instance 1 ethernet pw_evc
Router(config-if-srv)# encapsulation dot1q 2
Router(config_if-srv)# 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	"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*
Ethernet over MPLS	Ethernet over MPLS for the Cisco 7600 Series Internet Routers
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
Cisco IOS Carrier Ethernet commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Carrier Ethernet Command Reference*

Standards

Standard	Title
IEEE P802.1ag/D5.2	Draft Standard for Local and Metropolitan Area Networks
IETF VPLS OAM	*L2VPN OAM Requirements and Framework*
ITU-T	ITU-T Y.1731 OAM Mechanisms for Ethernet-Based Networks
ITU-T Q.3/13	Liaison statement on Ethernet OAM (Y.17ethoam)
Metro Ethernet Forum 16 Technical Specification	Technical Specification MEF 16-Ethernet Local Management Interface

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 Remote Port Shutdown

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 19 Feature Information for Configuring Remote Port Shutdown

Feature Name	Releases	Feature Information
Remote Port Shutdown	12.2(33)SRB	The Remote Port Shutdown feature uses Ethernet LMI in an EoMPLS network to propagate remote link status to a CE device. In Release 12.2(33)SRB, this feature was implemented on the Cisco 7600 router. The following commands were introduced or modified: **oam protocol**.

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 418

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 20 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**.

Configuring IEEE 802.3ad Link Bundling and Load Balancing

This document describes how the IEEE 802.3ad Link Bundling feature leverages the EtherChannel infrastructure within Cisco IOS software to manage the bundling of various links. Also described are network traffic load-balancing features to help minimize network disruption that results when a port is added or deleted from a link bundle.

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 802.3ad Link Bundling and Load Balancing

- Knowledge of how EtherChannels and Link Aggregation Control Protocol (LACP) function in a network
- Knowledge of load balancing to mitigate network traffic disruptions
- Verification that both ends of the LACP link have the same baseline software version

Restrictions for Configuring IEEE 802.3ad Link Bundling and Load Balancing

- The number of links supported per bundle is bound by the platform.
- On the Cisco 7600 series router, the maximum number of links per bundle is eight.
- On the Cisco 10000 series router, the maximum number of links per bundle is eight.
- On the Cisco 10000 series router only, 1-gigabit-per-second (Gbps) ports are supported for Gigabit EtherChannels (GECs).
- All links must operate at the same link speed and in full-duplex mode (LACP does not support half-duplex mode).
- All links must be configured either as EtherChannel links or as LACP links.
- Only physical interfaces can form aggregations. Aggregations of VLAN interfaces are not possible nor is an aggregation of aggregations.
- If a router is connected to a switch, the bundle terminates on the switch.
- An EtherChannel will not form if one of the LAN ports is a Switched Port Analyzer (SPAN) destination port.
- All ports in an EtherChannel must use the same EtherChannel protocol.
- LACP enhancements described in the LACP Enhancements Introduced in Cisco IOS Release 12.2(33)SB, page 434 are available only on the Cisco 10000 series router.
- The LACP Single Fault Direct Load Balance Swapping feature is limited to a single bundled port failure.
- The LACP Single Fault Direct Load Balance Swapping feature cannot be used with the Port Aggregation Protocol (PagP).
- LACP port priority cannot be configured with LACP single fault direct load balance swapping and vice versa.
- The adaptive algorithm does not apply to service control engines (SCEs) when EtherChannel load distribution is used.
- For the 802.3ad Link Aggregation with Weighted Load Balancing feature on the Cisco 7600 series router, the following maximum numbers of configurable service instances apply:
  - 8000 per port channel
  - 16,000 per line card
  - 64,000 per system
- The Cisco 7600 series router supports a maximum of 256 port channels.

Information About Configuring IEEE 802.3ad Link Bundling and Load Balancing

Gigabit EtherChannel

Gigabit EtherChannel is high-performance Ethernet technology that provides Gbps transmission rates. A Gigabit EtherChannel bundles individual Gigabit Ethernet links into a single logical link that provides the aggregate bandwidth of up to eight physical links. All LAN ports in each EtherChannel must be the same speed and all must be configured either as Layer 2 or as Layer 3 LAN ports. Inbound broadcast and multicast packets on one link in an EtherChannel are blocked from returning on any other link in the EtherChannel.

When a link within an EtherChannel fails, traffic previously carried over the failed link switches to the remaining links within that EtherChannel. Also when a failure occurs, a trap is sent that identifies the device, the EtherChannel, and the failed link.

Port Channel and LACP-Enabled Interfaces

Each EtherChannel has a numbered port channel interface that, if not already created, is created automatically when the first physical interface is added to the channel group. The configuration of a port channel interface affects all LAN ports assigned to that port channel interface.

To change the parameters of all ports in an EtherChannel, change the configuration of the port channel interface: for example, if you want to configure Spanning Tree Protocol or configure a Layer 2 EtherChannel as a trunk. Any configuration or attribute changes you make to the port channel interface are propagated to all interfaces within the same channel group as the port channel; that is, configuration changes are propagated to the physical interfaces that are not part of the port channel but are part of the channel group.

The configuration of a LAN port affects only that LAN port.

IEEE 802.3ad Link Bundling

The IEEE 802.3ad Link Bundling feature provides a method for aggregating multiple Ethernet links into a single logical channel based on the IEEE 802.3ad standard. This feature helps improve the cost effectiveness of a device by increasing cumulative bandwidth without necessarily requiring hardware upgrades. In addition, IEEE 802.3ad Link Bundling provides a capability to dynamically provision, manage, and monitor various aggregated links and enables interoperability between various Cisco devices and devices of third-party vendors.

LACP supports the automatic creation of EtherChannels by exchanging LACP packets between LAN ports. LACP packets are exchanged only between ports in passive and active modes. The protocol “learns” the capabilities of LAN port groups dynamically and informs the other LAN ports. After LACP identifies correctly matched Ethernet links, it facilitates grouping the links into an EtherChannel. Then the EtherChannel is added to the spanning tree as a single bridge port.

Both the passive and active modes allow LACP to negotiate between LAN ports to determine if they can form an EtherChannel, based on criteria such as port speed and trunking state. (Layer 2 EtherChannels also use VLAN numbers.) LAN ports can form an EtherChannel when they are in compatible LACP modes, as in the following examples:

- A LAN port in active mode can form an EtherChannel with another LAN port that is in active mode.
- A LAN port in active mode can form an EtherChannel with another LAN port that is in passive mode.

- A LAN port in passive mode cannot form an EtherChannel with another LAN port that is also in passive mode because neither port will initiate negotiation.

LACP uses the following parameters:

- LACP system priority--You must configure an LACP system priority on each device running LACP. The system priority can be configured automatically or through the command-line interface (CLI). LACP uses the system priority with the device MAC address to form the system ID and also during negotiation with other systems.
- LACP port priority--You must configure an LACP port priority on each port configured to use LACP. The port priority can be configured automatically or through the CLI. LACP uses the port priority to decide which ports should be put in standby mode when there is a hardware limitation that prevents all compatible ports from aggregating. LACP also uses the port priority with the port number to form the port identifier.
- LACP administrative key--LACP automatically configures an administrative key value on each port configured to use LACP. The administrative key defines the ability of a port to aggregate with other ports. A port's ability to aggregate with other ports is determined by the following:
  - Port physical characteristics such as data rate, duplex capability, and point-to-point or shared medium
  - Configuration restrictions that you establish

LACP, on ports configured to use it, tries to configure the maximum number of compatible ports in an EtherChannel, up to the maximum allowed by the hardware. In Cisco IOS Release 12.2(31)SB2 on the Cisco 10000 series router, only four ports per bundle can be aggregated and the peer must be configured to support LACP. To use the hot standby feature in the event a channel port fails, both ends of the LACP bundle must support the **lacp max-bundle** command.

As a control protocol, LACP uses the Slow Protocol multicast address of 01-80-C2-00-00-02 to transmit LACP protocol data units (PDUs). Operations, administration, and maintenance (OAM) packets also use the Slow Protocol link type. Subsequently, a subtype field is defined per the IEEE 802.3ad standard [1] (Annex 43B, section 4) differentiating LACP PDUs from OAM PDUs.

- Benefits of IEEE 802.3ad Link Bundling, page 434

Benefits of IEEE 802.3ad Link Bundling

- Increased network capacity without changing physical connections or upgrading hardware
- Cost savings resulting from use of existing hardware and software for additional functions
- A standard solution that enables interoperability of network devices
- Port redundancy without user intervention when an operational port fails

LACP Enhancements Introduced in Cisco IOS Release 12.2(33)SB

In Cisco IOS Release 12.2(33)SB on the Cisco 10000 series router, the following LACP enhancements are supported:

- Eight member links per LACP bundle.
- Stateful switchover (SSO), In Service Software Upgrade (ISSU), Cisco nonstop forwarding (NSF), and nonstop routing (NSR) on Gigabit EtherChannel bundles.
- Point-to-Point Protocol over Ethernet over Ethernet (PPPoEoE), Point-to-Point Protocol over Ethernet over IEEE 802.1Q in 802.1Q (PPPoEoQinQ), and Point-to-Point Protocol over VLAN (PPPoVLAN) sessions are not forced to reestablish when a link switchover occurs. During the switchover, the port

channel is maintained in the LINK_UP state, and both the active and standby links assume the same configured elements after the switchover.

- Link failover time of 250 milliseconds or less and a maximum link failover time of 2 seconds; port channels remain in the LINK_UP state to eliminate reconvergence by the Spanning-Tree Protocol.
- Shutting down a port channel when the number of active links falls below the minimum threshold. In the port channel interface, a configurable option is provided to bring down the port channel interface when the number of active links falls below the minimum threshold. For the port-channel state to be symmetric on both sides of the channel, the peer must also be running LACP and have the same **lacp min-bundle** command setting.
- The IEEE LAG MIB.

EtherChannel Load Balancing

EtherChannel load balancing can use MAC addresses; IP addresses; Layer 4 port numbers; either source addresses, destination addresses, or both; or ports. The selected mode applies to all EtherChannels configured on the device. EtherChannel load balancing can also use Multiprotocol Label Switching (MPLS) Layer 2 information.

Traffic load across the links in an EtherChannel is balanced by reducing part of the binary pattern, formed from the addresses in the frame, to a numerical value that selects one of the links in the channel. When a port is added to an EtherChannel or an active port fails, the load balance bits are reset and reassigned for all ports within that EtherChannel and reprogrammed into the ASIC for each port. This reset causes packet loss during the time the reassignment and reprogramming is taking place. The greater the port bandwidth, the greater the packet loss.

LACP Single Fault Direct Load Balance Swapping

LACP supports hot standby ports, which are created when a platform's maximum number of ports that can be aggregated are bundled. On the Cisco 7600 router, eight is the maximum number of ports that can be bundled. A hot standby port is bundled in (swapped into) an aggregation when a previously active port fails.

The LACP Single Fault Direct Load Balance Swapping feature reassigns the load balance bits so that the swapped-in hot standby port is assigned the load balance bits of the failed port, and the load balance bits of the remaining ports in the aggregation remain unchanged. When the swapped-in port is bundled, the stored load share of the failed port is assigned to the swapped-in port. The remaining ports in the bundle are not affected.

The LACP Single Fault Direct Load Balance Swapping feature addresses a single bundled port failure. If a second failure occurs before the first failure recovers, the load share bits for member links are recomputed.

Following is an overview of the LACP single fault direct load balance swapping process:

1. When a failed (unbundled) port is detected and is the first failure, its load share is stored.
2. When a hot-standby port is identified and is bundled in, it takes the load share bits of the previously failed port.
3. If the failed port comes back up, it replaces the hot-standby port in the bundle and the load share bits are transferred back to the original port.

The LACP Single Fault Direct Load Balance Swapping feature is enabled using the CLI command **lacp direct-loadswap** in port-channel configuration mode.

Load Distribution in an EtherChannel

Prior to Cisco IOS Release 12.(33)SRC, only a fixed load distribution algorithm was supported. With this fixed algorithm, the load share bits are assigned sequentially to each port in the bundle. Consequently, the load share bits for existing ports change when a member link joins or leaves the bundle. When these values are programmed in the ASIC, substantial traffic disruption and, in some cases, duplication of traffic can occur.

The Load Distribution in an EtherChannel feature enhances the load distribution mechanism with the adaptive load distribution algorithm. This algorithm uses a port reassignment scheme that enhances EtherChannel availability by limiting the load distribution reassignment to the port that is added or deleted. The new load on existing bundled ports does not conflict with the load programmed on those ports when a port is added or deleted.

You can enable this feature in either global configuration mode or interface configuration mode. The algorithm is applied at the next hash-distribution instance, which usually occurs when a link fails, is activated, added, or removed, or when shutdown or no shutdown is configured.

Because the selected algorithm is not applied until the next hash-distribution instance, the current and configured algorithms could be different. If the algorithms are different, a message is displayed alerting you to take appropriate action. For example:

```
Router(config-if)# port-channel port hash-distribution
fixed
This command will take effect upon a member link UP/DOWN/ADDITION/DELETION event.
Please do a shut/no shut to take immediate effect
```

Also, the output of the **show etherchannel** command is enhanced to show the applied algorithm when the channel group number is specified. This output enhancement is not available, though, when the protocol is also specified because only protocol-specific information is included. Following is an example of output showing the applied algorithm:

```
Router# show etherchannel
 10 summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      N - not in use, no aggregation
        f - failed to allocate aggregator
<snip>
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
10     Po10(RU)        LACP      Gi3/7(P)       Gi3/9(P)
! The following line of output is added with support
of the EtherChannel Load Distribution feature. !
Last applied Hash Distribution Algorithm: Fixed
```

802.3ad Link Aggregation with Weighted Load Balancing

Current mechanisms for load balancing Ethernet service instances over member links in a port channel do not account for the service instances' traffic loads, which can lead to unequal distribution of traffic over member links. The 802.3ad Link Aggregation with Weighted Load Balancing feature (802.3ad LAG with WLB) is an enhancement introduced in Cisco IOS Release 15.0(1)S that allows you to assign weights to service instances to efficiently distribute traffic flow across active member links in a port channel.

The LAG with WLB feature supports both LACP (active or passive mode) and manual (mode on) EtherChannel bundling. A weighted load balancing configuration does not affect the selection of active

member links in the EtherChannel. As member links become active or inactive, a load-balancing algorithm adjusts the distribution of Ethernet service instances to use the currently active member links.

- Load Balancing Coexistence, page 437
- Service Group Support, page 437

Load Balancing Coexistence

With the added support for weighted load balancing, three methods for load balancing Ethernet service instances over port-channel member links are available. The method used is selected in the following order (highest precedence first):

1. Manual load balancing
2. Weighted load balancing
3. Platform default load balancing

If an Ethernet service instance is configured to be manually assigned to a member link and that member link is an active member of the port channel, that manual assignment is applied. If the Ethernet service instance is not manually load balanced and weighted load balancing is enabled with the **port-channel load-balance weighted link** command, the service instance is load balanced based on its configured or default weight. If neither the manual nor weighted method is applied to the service instance, the platform default load-balancing mechanism is used.

When both manual and weighted methods are load balancing Ethernet service instances over the same member link or links, the weights of the manually load-balanced service instances are included in determining weight distributions. As with every other Ethernet service instance, if a weight is not specifically configured on a manually load-balanced Ethernet service instance, the default weight is used.

The weighted load balancing method can be configured to use only a specific number of member links. This configuration option allows one or more member links to be dedicated to the manually load-balanced Ethernet service instances.

Service Group Support

An Ethernet service group is a logical collection of Ethernet service instances, subinterfaces, or both. Traffic for all Ethernet service instances that are members of a service group must egress the same member link. This restriction is necessary for quality of service (QoS) configured for the service group to perform accurate computations but could lead to unequal weight distributions across the available member links. For example, consider 100 Ethernet service instances in a service group, each configured with a weight of 1, and one other Ethernet service instance configured with a weight of 2 that is not in a service group. In this case, one member link will have a total weight of 100 and another member link will have a total weight of 2. This example is not a typical scenario but illustrates the traffic imbalance that could result.

How to Configure IEEE 802.3ad Link Bundling and Load Balancing

- Enabling LACP, page 438
- Configuring a Port Channel, page 439
- Associating a Channel Group with a Port Channel, page 440
- Setting LACP System Priority, page 442

Enabling LACP

Perform this task to enable LACP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface port-channel** *channel-number*
4. **channel-group** *channel-group-number* **mode** {**active** | **passive**}
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 port-channel** *channel-number* **Example:** `Router(config)# interface port-channel 10`	Identifies the interface port channel and enters interface configuration mode.
Step 4	**channel-group** *channel-group-number* **mode** {**active** | **passive**} **Example:** `Router(config-if)# channel-group 25 mode active`	Configures the interface in a channel group and sets it as active. • In active mode, the port will initiate negotiations with other ports by sending LACP packets.

	Command or Action	Purpose
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuring a Port Channel

You must manually create a port channel logical interface. Perform this task to configure a port channel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface port-channel** *channel-number*
4. **ip address** *ip-address mask*
5. **end**
6. **show running-config interface port-channel** *group-number*
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	**interface port-channel** *channel-number* **Example:** `Router(config)# interface port-channel 10`	Identifies the interface port channel and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip address** *ip-address mask* **Example:** `Router(config-if)# ip address 172.31.52.10 255.255.255.0`	Assigns an IP address and subnet mask to the EtherChannel.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.
Step 6	**show running-config interface port-channel** *group-number* **Example:** `Router# show running-config interface port-channel 10`	Displays the port channel configuration.
Step 7	**end** **Example:** `Router# end`	Ends the current configuration session.

Example

This example shows how to verify the configuration:

```
Router# show running-config interface port-channel10

Building configuration...
Current configuration:
!
interface Port-channel10
 ip address 172.31.52.10 255.255.255.0
 no ip directed-broadcast
end
```

Associating a Channel Group with a Port Channel

Perform this task to associate a channel group with a port channel.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface port-channel** *channel -number*
4. **interface** *type number*
5. **channel-group** *channel-group-number* **mode** {**active** | **passive**}
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 port-channel** *channel -number* **Example:** `Router(config)# interface port-channel 5`	Creates a port channel.
Step 4	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet 7/0/0`	Configures an interface and enters interface configuration mode.
Step 5	**channel-group** *channel-group-number* **mode** {**active**	**passive**} **Example:** `Router(config-if)# channel-group 5 mode active`	Includes the interface as part of the port channel bundle.

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

Setting LACP System Priority

Perform this task to set the LACP system priority. The system ID is the combination of the LACP system priority and the MAC address of a device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **lacp system-priority** *priority*
4. **end**
5. **show lacp sys-id**
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	**lacp system-priority** *priority* **Example:** `Router(config)# lacp system-priority 200`	Sets the system priority.

	Command or Action	Purpose
Step 4	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.
Step 5	**show lacp sys-id** **Example:** `Router# show lacp`	Displays the system ID, which is a combination of the system priority and the MAC address of the device.
Step 6	**end** **Example:** `Router# end`	Ends the current configuration session.

Example

This example shows how to verify the LACP configuration:

```
Router# show lacp
20369,01b2.05ab.ccd0
```

Adding and Removing Interfaces from a Bundle

Perform this task to add and remove an interface from a link bundle.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **channel-group** *channel-group-number* **mode** {**active** | **passive**}
5. **no channel-group**
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 gigabitethernet 5/0/0`	Configures an interface and enters interface configuration mode.
Step 4	**channel-group** *channel-group-number* **mode** {**active** | **passive**} **Example:** `Router(config-if)# channel-group 5 mode active`	Adds an interface to a channel group.
Step 5	**no channel-group** **Example:** `Router(config-if)# no channel-group`	Removes the interface from the channel group.
Step 6	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Setting a Minimum Number of Active Links

Perform this task to set the minimum number of active links allowed in an LACP bundle.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **lacp min-bundle** *min-bundle*
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 port-channel 1`	Creates a port-channel virtual interface and enters interface configuration mode.
Step 4	**lacp min-bundle** *min-bundle* **Example:** `Router(config-if)# lacp min-bundle 5`	Sets the minimum threshold of active links.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Monitoring LACP Status

Perform this task to monitor LACP activity in the network.

SUMMARY STEPS

1. **enable**
2. **show lacp** {*number* | **counters** | **internal** | **neighbor** | **sys-id**}
3. **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 lacp** {*number*	**counters**	**internal**	**neighbor**	**sys-id**} **Example:** `Router# show lacp internal`	Displays internal device information.
Step 3	**end** **Example:** `Router# end`	Ends the current configuration session.

• Troubleshooting Tips, page 446

Troubleshooting Tips

Use the **debug lacp** command to display LACP configuration and activity details.

The following sample output from a **debug lacp all** command shows that a remote device is removing a link and also adding a link:

```
Router# debug lacp all
Link Aggregation Control Protocol all debugging is on
Router#
*Aug 20 17:21:51.685: LACP :lacp_bugpak: Receive LACP-PDU packet via Gi5/0/0
*Aug 20 17:21:51.685: LACP : packet size: 124
*Aug 20 17:21:51.685: LACP: pdu: subtype: 1, version: 1
*Aug 20 17:21:51.685: LACP: Act: tlv:1, tlv-len:20, key:0x1, p-pri:0x8000, p:0x14, p-
state:0x3C,
s-pri:0xFFFF, s-mac:0011.2026.7300
*Aug 20 17:21:51.685: LACP: Part: tlv:2, tlv-len:20, key:0x5, p-pri:0x8000, p:0x42, p-
state:0x3D,
s-pri:0x8000, s-mac:0014.a93d.4a00
*Aug 20 17:21:51.685: LACP: col-tlv:3, col-tlv-len:16, col-max-d:0x8000
*Aug 20 17:21:51.685: LACP: term-tlv:0 termr-tlv-len:0
*Aug 20 17:21:51.685: LACP: Gi5/0/0 LACP packet received, processing
*Aug 20 17:21:51.685:     lacp_rx Gi5: during state CURRENT, got event 5(recv_lacpdu)
*Aug 20 17:21:59.869: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:21:59.869: LACP: lacp_p(Gi5/0/0) expired
```

```
*Aug 20 17:21:59.869:     lacp_ptx Gi5: during state SLOW_PERIODIC, got event
3(pt_expired)
*Aug 20 17:21:59.869: @@@ lacp_ptx Gi5: SLOW_PERIODIC -> PERIODIC_TX
*Aug 20 17:21:59.869: LACP: Gi5/0/0 lacp_action_ptx_slow_periodic_exit entered
*Aug 20 17:21:59.869: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:22:00.869: LACP: lacp_t(Gi5/0/0) timer stopped
*Aug 20 17:22:00.869: LACP: lacp_t(Gi5/0/0) expired
*Aug 20 17:22:19.089: LACP :lacp_bugpak: Receive LACP-PDU packet via Gi5/0/0
*Aug 20 17:22:19.089: LACP : packet size: 124
*Aug 20 17:22:19.089: LACP: pdu: subtype: 1, version: 1
*Aug 20 17:22:19.089: LACP: Act: tlv:1, tlv-len:20, key:0x1, p-pri:0x8000, p:0x14, p-
state:0x4,
s-pri:0xFFFF, s-mac:0011.2026.7300
*Aug 20 17:22:19.089: LACP: Part: tlv:2, tlv-len:20, key:0x5, p-pri:0x8000, p:0x42, p-
state:0x34,
s-pri:0x8000, s-mac:0014.a93d.4a00
*Aug 20 17:22:19.089: LACP: col-tlv:3, col-tlv-len:16, col-max-d:0x8000
*Aug 20 17:22:19.089: LACP: term-tlv:0 termr-tlv-len:0
*Aug 20 17:22:19.089: LACP: Gi5/0/0 LACP packet received, processing
*Aug 20 17:22:19.089:     lacp_rx Gi5: during state CURRENT, got event 5(recv_lacpdu)
*Aug 20 17:22:19.989: LACP: lacp_t(Gi5/0/0) timer stopped
*Aug 20 17:22:19.989: LACP: lacp_t(Gi5/0/0) expired
*Aug 20 17:22:19.989: LACP: timer lacp_t(Gi5/0/0) started with interval 1000.
*Aug 20 17:22:19.989: LACP: lacp_send_lacpdu: (Gi5/0/0) About to send the 110 LACPDU
*Aug 20 17:22:19.989: LACP :lacp_bugpak: Send LACP-PDU packet via Gi5/0/0
*Aug 20 17:22:19.989: LACP : packet size: 124
*Aug 20 17:22:20.957: LACP: lacp_t(Gi5/0/0) timer stopped
*Aug 20 17:22:20.957: LACP: lacp_t(Gi5/0/0) expired
*Aug 20 17:22:21.205: %LINK-3-UPDOWN: Interface GigabitEthernet5/0/0, changed state to
down
*Aug 20 17:22:21.205: LACP: lacp_hw_off: Gi5/0/0 is going down
*Aug 20 17:22:21.205: LACP: if_down: Gi5/0/0
*Aug 20 17:22:21.205:     lacp_ptx Gi5: during state SLOW_PERIODIC, got event
0(no_periodic)
*Aug 20 17:22:22.089: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel5,
changed state to down
*Aug 20 17:22:22.153: %C10K_ALARM-6-INFO: CLEAR CRITICAL GigE 5/0/0 Physical Port Link
Down
*Aug 20 17:22:23.413: LACP: Gi5/0/0 oper-key: 0x0
*Aug 20 17:22:23.413: LACP: lacp_hw_on: Gi5/0/0 is coming up
*Aug 20 17:22:23.413:     lacp_ptx Gi5: during state NO_PERIODIC, got event 0(no_periodic)
*Aug 20 17:22:23.413: @@@ lacp_ptx Gi5: NO_PERIODIC -> NO_PERIODIC
*Aug 20 17:22:23.413: LACP: Gi5/0/0 lacp_action_ptx_no_periodic entered
*Aug 20 17:22:23.413: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:22:24.153: %LINK-3-UPDOWN: Interface GigabitEthernet5/0/0, changed state to up
*Aug 20 17:22:24.153: LACP: lacp_hw_on: Gi5/0/0 is coming up
*Aug 20 17:22:24.153:     lacp_ptx Gi5: during state FAST_PERIODIC, got event
0(no_periodic)
*Aug 20 17:22:24.153: @@@ lacp_ptx Gi5: FAST_PERIODIC -> NO_PERIODIC
*Aug 20 17:22:24.153: LACP: Gi5/0/0 lacp_action_ptx_fast_periodic_exit entered
*Aug 20 17:22:24.153: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:22:24.153: LACP:
*Aug 20 17:22:25.021: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:22:25.021: LACP: lacp_p(Gi5/0/0) expired
*Aug 20 17:22:25.021:     lacp_ptx Gi5: during state FAST_PERIODIC, got event
3(pt_expired)
*Aug 20 17:22:25.021: @@@ lacp_ptx Gi5: FAST_PERIODIC -> PERIODIC_TX
*Aug 20 17:22:25.021: LACP: Gi5/0/0 lacp_action_ptx_fast_periodic_exit entered
*Aug 20 17:22:25.021: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:22:25.917: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:22:25.917: LACP: lacp_p(Gi5/0/0) expired
*Aug 20 17:22:25.917:     lacp_ptx Gi5: during state FAST_PERIODIC, got event
3(pt_expired)
*Aug 20 17:22:25.917: @@@ lacp_ptx Gi5: FAST_PERIODIC -> PERIODIC_TX
*Aug 20 17:22:25.917: LACP: Gi5/0/0 lacp_action_ptx_fast_periodic_exit entered
*Aug 20 17:22:25.917: LACP: lacp_p(Gi5/0/0) timer stopped
```

The following sample output shows a remote device adding a link:

```
Router#
*Aug 20 17:23:54.005: LACP: lacp_t(Gi5/0/0) timer stopped
*Aug 20 17:23:54.005: LACP: lacp_t(Gi5/0/0) expired
*Aug 20 17:23:55.789: %C10K_ALARM-6-INFO: ASSERT CRITICAL GigE 5/0/0 Physical Port Link
```

```
Down
*Aug 20 17:23:56.497: %C10K_ALARM-6-INFO: CLEAR CRITICAL GigE 5/0/0 Physical Port Link
Down
*Aug 20 17:24:19.085: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:24:19.085: LACP: lacp_p(Gi5/0/0) expired
*Aug 20 17:24:19.085:     lacp_ptx Gi5: during state SLOW_PERIODIC, got event
3(pt_expired)
*Aug 20 17:24:19.085: @@@ lacp_ptx Gi5: SLOW_PERIODIC -> PERIODIC_TX
*Aug 20 17:24:19.085: LACP: Gi5/0/0 lacp_action_ptx_slow_periodic_exit entered
*Aug 20 17:24:19.085: LACP: lacp_p(Gi5/0/0) timer stopped
*Aug 20 17:24:19.957: LACP: lacp_t(Gi5/0/0) timer stopped
*Aug 20 17:24:19.957: LACP: lacp_t(Gi5/0/0) expired
*Aug 20 17:24:21.073: LACP :lacp_bugpak: Receive LACP-PDU packet via Gi5/0/0
*Aug 20 17:24:21.073: LACP : packet size: 124
*Aug 20 17:24:21.073: LACP: pdu: subtype: 1, version: 1
*Aug 20 17:24:21.073: LACP: Act: tlv:1, tlv-len:20, key:0x1, p-pri:0x8000, p:0x14, p-
state:0xC,
s-pri:0xFFFF, s-mac:0011.2026.7300
*Aug 20 17:24:21.073: LACP: Part: tlv:2, tlv-len:20, key:0x0, p-pri:0x8000, p:0x42, p-
state:0x75,
s-pri:0x8000, s-mac:0014.a93d.4a00
*Aug 20 17:24:21.073: LACP: col-tlv:3, col-tlv-len:16, col-max-d:0x8000
*Aug 20 17:24:21.073: LACP: term-tlv:0 termr-tlv-len:0
*Aug 20 17:24:21.073: LACP: Gi5/0/0 LACP packet received, processing
*Aug 20 17:24:21.073:     lacp_rx Gi5: during state DEFAULTED, got event 5(recv_lacpdu)
*Aug 20 17:24:21.929: LACP: lacp_t(Gi5/0/0) timer stopped
*Aug 20 17:24:21.929: LACP: lacp_t(Gi5/0/0) expired
*Aug 20 17:24:21.929: LACP: timer lacp_t(Gi5/0/0) started with interval 1000.
*Aug 20 17:24:21.929: LACP: lacp_send_lacpdu: (Gi5/0/0) About to send the 110 LACPDU
*Aug 20 17:24:21.929: LACP :lacp_bugpak: Send LACP-PDU packet via Gi5/0/0
*Aug 20 17:24:21.929: LACP : packet size: 124
*Aug 20 17:24:22.805: LACP: lacp_t(Gi5/0/0) timer stopped
*Aug 20 17:24:22.805: LACP: lacp_t(Gi5/0/0) expired
*Aug 20 17:24:23.025: LACP: lacp_w(Gi5/0/0) timer stopped
*Aug 20 17:24:23.025: LACP: lacp_w(Gi5/0/0) expired
*Aug 20 17:24:23.025:     lacp_mux Gi5: during state WAITING, got event 4(ready)
*Aug 20 17:24:23.025: @@@ lacp_mux Gi5: WAITING -> ATTACHED
*Aug 20 17:24:23.921: LACP: lacp_t(Gi5/0/0) timer stopped
*Aug 20 17:24:23.921: LACP: lacp_t(Gi5/0/0) expired
*Aug 20 17:24:26.025: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel5,
changed state to up
```

Enabling LACP Single Fault Load Balance Swapping

Perform this task to enable LACP single fault load balance swapping in EtherChannels.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **lacp direct-loadswap**
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 port-channel 1`	Creates a port-channel virtual interface and enters interface configuration mode.
Step 4	**lacp direct-loadswap** **Example:** `Router(config-if)# lacp direct-loadswap`	Enables LACP single fault direct load balancing.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Selecting an EtherChannel Load Distribution Algorithm

You can select the EtherChannel load distribution algorithm from either global configuration mode or interface configuration mode. Perform this task to select either the adaptive or fixed algorithm from global configuration mode. To select the algorithm from interface configuration mode, issue the **interface** command before the **port-channel hash-distribution** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **port-channel hash-distribution** {**adaptive** | **fixed**}
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 port-channel1`	(Optional) Creates a port-channel virtual interface and enters interface configuration mode.
Step 4	**port-channel hash-distribution** {**adaptive** | **fixed**} **Example:** `Router(config)# port-channel hash-distribution adaptive`	Selects the type of algorithm. **Note** If an algorithm is not specified in interface configuration mode, the global configuration is applied. Otherwise, the algorithm specified in interface configuration mode overrides the algorithm specified in global configuration mode.
Step 5	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Enabling 802.3ad Weighted Load Balancing

Perform this task to enable 802.3ad weighted load balancing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **port-channel load-balance** {**link** *link-id* | **weighted** {**default weight** *weight* | **link** {**all** | *link-id*} | **rebalance**{**disable** | *weight*}}}
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 portchannel10`	Configures a port-channel interface and enters interface configuration mode.
Step 4	**port-channel load-balance** {**link** *link-id* | **weighted** {**default weight** *weight* | **link** {**all** | *link-id*} | **rebalance**{**disable** | *weight*}}} **Example:** `Router(config-if)# port-channel load-balance weighted link all`	Configures weighted load balancing on port-channel member links.
Step 5	**end** **Example:** `Router(config-if)# end`	Returns to privileged EXEC mode.

Configuration Examples for Configuring IEEE 802.3ad Link Bundling and Load Balancing

Example Associating a Channel Group with a Port Channel

This example shows how to configure channel group number 5 and include it in the channel group:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface port-channel5
Router(config-if)#
*Aug 20 17:06:14.417: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel5,
changed state to down
*Aug 20 17:06:25.413: %LINK-3-UPDOWN: Interface Port-channel5, changed state to down
Router(config-if)#
Router(config-if)# interface gigabitethernet 7/0/0
Router(config-if)# channel-group 5 mode active
Router(config-if)#
*Aug 20 17:07:43.713: %LINK-3-UPDOWN: Interface GigabitEthernet7/0/0, changed state to
down
*Aug 20 17:07:44.713: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet7/0/0, changed state to down
*Aug 20 17:07:45.093: %C10K_ALARM-6-INFO: ASSERT CRITICAL GigE 7/0/0 Physical Port Link
Down
*Aug 20 17:07:45.093: %C10K_ALARM-6-INFO: CLEAR CRITICAL GigE 7/0/0 Physical Port Link
Down
*Aug 20 17:07:47.093: %LINK-3-UPDOWN: Interface GigabitEthernet7/0/0, changed state to up
*Aug 20 17:07:48.093: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet7/0/0, changed state to up
*Aug 20 17:07:48.957: GigabitEthernet7/0/0 added as member-1 to port-channel5

*Aug 20 17:07:51.957: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel5,
changed state to up
Router(config-if)# end
Router#
*Aug 20 17:08:00.933: %SYS-5-CONFIG_I: Configured from console by console
Router# show lacp internal
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode
Channel group 5
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Gi7/0/0   SA      bndl      32768         0x5       0x5     0x43        0x3D
Router# show interface port-channel5
Port-channel5 is up, line protocol is up
  Hardware is GEChannel, address is 0014.a93d.4aa8 (bia 0000.0000.0000)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
    No. of active members in this channel: 1
        Member 0 : GigabitEthernet7/0/0 , Full-duplex, 1000Mb/s
  Last input 00:00:05, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Interface Port-channel5 queueing strategy: PXF First-In-First-Out
  Output queue 0/8192, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     9 packets output, 924 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
```

Example Adding and Removing Interfaces from a Bundle

The following example shows how to add an interface to a bundle:

```
Router# show lacp internal
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode
Channel group 5
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Gi7/0/0   SA      bndl      32768         0x5       0x5     0x43        0x3D
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 5/0/0
Router(config-if)# channel-group 5 mode active
Router(config-if)#
*Aug 20 17:10:19.057: %LINK-3-UPDOWN: Interface GigabitEthernet5/0/0, changed state to
down
*Aug 20 17:10:19.469: %C10K_ALARM-6-INFO: ASSERT CRITICAL GigE 5/0/0 Physical Port Link
Down
*Aug 20 17:10:19.473: %C10K_ALARM-6-INFO: CLEAR CRITICAL GigE 5/0/0 Physical Port Link
Down
*Aug 20 17:10:21.473: %LINK-3-UPDOWN: Interface GigabitEthernet5/0/0, changed state to up
*Aug 20 17:10:21.473: GigabitEthernet7/0/0 taken out of port-channel5
*Aug 20 17:10:23.413: GigabitEthernet5/0/0 added as member-1 to port-channel5

*Aug 20 17:10:23.473: %LINK-3-UPDOWN: Interface Port-channel5, changed state to up
Router(config-if)# end
Router#
*Aug 20 17:10:27.653: %SYS-5-CONFIG_I: Configured from console by console
*Aug 20 17:11:40.717: GigabitEthernet7/0/0 added as member-2 to port-channel5

Router# show lacp internal
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode
Channel group 5
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Gi7/0/0   SA      bndl      32768         0x5       0x5     0x43        0x3D
Gi5/0/0   SA      bndl      32768         0x5       0x5     0x42        0x3D
Router# show interface port-channel5
Port-channel5 is up, line protocol is up
  Hardware is GEChannel, address is 0014.a93d.4aa8 (bia 0000.0000.0000)
  MTU 1500 bytes, BW 2000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
    No. of active members in this channel: 2
        Member 0 : GigabitEthernet5/0/0 , Full-duplex, 1000Mb/s  <---- added to port
channel bundle
        Member 1 : GigabitEthernet7/0/0 , Full-duplex, 1000Mb/s
  Last input 00:00:00, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/150/0/0 (size/max/drops/flushes); Total output drops: 0
  Interface Port-channel5 queueing strategy: PXF First-In-First-Out
  Output queue 0/8192, 0 drops; input queue 0/150, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast, 0 pause input
     104 packets output, 8544 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
```

```
     0 lost carrier, 0 no carrier, 0 PAUSE output
     0 output buffer failures, 0 output buffers swapped out
```

The following example shows how to remove an interface from a bundle:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface gigabitethernet 7/0/0
Router(config-if)# no channel-group
Router(config-if)#
*Aug 20 17:15:49.433: GigabitEthernet7/0/0 taken out of port-channel5
*Aug 20 17:15:49.557: %C10K_ALARM-6-INFO: ASSERT CRITICAL GigE 5/0/0 Physical Port Link
Down
*Aug 20 17:15:50.161: %C10K_ALARM-6-INFO: CLEAR CRITICAL GigE 5/0/0 Physical Port Link
Down
*Aug 20 17:15:51.433: %LINK-3-UPDOWN: Interface GigabitEthernet7/0/0, changed state to
down
*Aug 20 17:15:52.433: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet7/0/0, changed state to down
Router(config-if)# end
Router#
*Aug 20 17:15:58.209: %SYS-5-CONFIG_I: Configured from console by console
Router#
*Aug 20 17:15:59.257: %C10K_ALARM-6-INFO: ASSERT CRITICAL GigE 7/0/0 Physical Port Link
Down
*Aug 20 17:15:59.257: %C10K_ALARM-6-INFO: CLEAR CRITICAL GigE 7/0/0 Physical Port Link
Down
Router#
*Aug 20 17:16:01.257: %LINK-3-UPDOWN: Interface GigabitEthernet7/0/0, changed state to up
*Aug 20 17:16:02.257: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet7/0/0, changed state to up
Router# show lacp internal
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode
Channel group 5
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Gi5/0/0   SA      bndl      32768         0x5       0x5     0x42        0x3D
```

Example Monitoring LACP Status

The following example shows LACP activity that you can monitor by using the **show lacp** command.

```
Router# show lacp internal
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode
Channel group 5
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Gi5/0/0   SA      bndl      32768         0x5       0x5     0x42        0x3D
Router# show lacp 5 counters
           LACPDUs         Marker      Marker Response    LACPDUs
Port       Sent   Recv     Sent   Recv     Sent   Recv      Pkts Err
---------------------------------------------------------------------
Channel group: 5
Gi5/0/0     21     18       0      0        0      0         0
Router# show lacp 5 internal
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode
Channel group 5
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Gi5/0/0   SA      bndl      32768         0x5       0x5     0x42        0x3D
Router# show lacp 5 neighbor
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode
```

```
Channel group 5 neighbors
Partner's information:
          Partner Partner   LACP Partner  Partner   Partner  Partner     Partner
Port      Flags   State     Port Priority Admin Key Oper Key Port Number Port State
Gi5/0/0   SP      32768     0011.2026.7300  11s    0x1     0x14     0x3C
Router# show lacp counters
            LACPDUs          Marker      Marker Response    LACPDUs
Port       Sent   Recv     Sent   Recv     Sent   Recv      Pkts Err
---------------------------------------------------------------------
Channel group: 5
Gi5/0/0     23     20       0      0        0      0         0
Router# s
how lacp sys-id
32768,0014.a93d.4a00
```

Example Configuring Weighted Service Instances

In this example traffic on service instances 100, 101, and 200 is load balanced over Gigabit Ethernet interfaces 5/0/2 and 5/0/3. Based on the configured weights, traffic from service instances 100 and 101 egress one member link, and traffic from service instance 200 egress the other member link.

```
Router# configure terminal
Router(config)# interface GigabitEthernet5/0/2
Router(config-if)# channel-group 10 mode on
Router(config-if)# exit
Router(config)# interface GigabitEthernet5/0/3
Router(config-if)# channel-group 10 mode on
Router(config-if)# exit
Router(config)# interface Port-channel10
Router(config-if)# port-channel load-balance weighted link all
Router(config-if)# service instance 100 ethernet
Router(config-if-srv)# encapsulation dot1q 100
Router(config-if-srv)# weight 2
Router(config-if-srv)# exit
Router(config-if)# service instance 101 ethernet
Router(config-if-srv)# encapsulation dot1q 101
Router(config-if-srv)# weight 2
Router(config-if-srv)# exit
Router(config-if)# service instance 200 ethernet
Router(config-if-srv)# encapsulation dot1q 200
Router(config-if-srv)# weight 10
Router(config-if-srv)# end
```

Example Configuring Weighted and Manual Load Balancing

In this example a combination of manual load balancing and weighted load balancing is configured. Service instances 100 and 101 are manually assigned to link 1 on Gigabit Ethernet interface 5/0/2. Both link 2 on Gigabit Ethernet interface 5/0/3 and link 3 on Gigabit Ethernet interface 5/0/4 are configured for weighted load balancing. Because service instances 200 and 201 are not configured with explicit weights, they inherit the configured default of 2. Service instances 200, 201, and 300 are distributed across Gigabit Ethernet interfaces 5/0/3 and 5/0/4.

```
Router(config)# interface GigabitEthernet5/0/2
Router(config-if)# channel-group 10 mode on link 1
Router(config-if)# exit
Router(config)# interface GigabitEthernet5/0/3
Router(config-if)# channel-group 10 mode on link 2
Router(config-if)# exit
Router(config)# interface GigabitEthernet5/0/4
Router(config-if)# channel-group 10 mode on link 3
Router(config-if)# exit
!
Router(config)# interface Port-channel10
Router(config-if)# port-channel load-balance link 1
Router(config-if)# service-instance 100-150
```

```
Router(config-if)# port-channel load-balance weighted link 2,3
Router(config-if)# port-channel load-balance weighted default weight 2
Router(config-if)# port-channel load-balance weighted rebalance disable
Router(config-if)# service instance 100 ethernet
Router(config-if-srv)# encapsulation dot1q 100
Router(config-if-srv)# exit
Router(config-if)# service instance 101 ethernet
Router(config-if-srv)# encapsulation dot1q 101
Router(config-if-srv)# exit
Router(config-if)# service instance 200 ethernet
Router(config-if-srv)# encapsulation dot1q 200
Router(config-if-srv)# exit
Router(config-if)# service instance 201 ethernet
Router(config-if-srv)# encapsulation dot1q 201
Router(config-if-srv)# exit
Router(config-if)# service instance 300 ethernet
Router(config-if-srv)# encapsulation dot1q 300
Router(config-if-srv)# weight 5
Router(config-if-srv)# end
```

Additional References

Related Documents

Related Topic	Document Title
Configuring EtherChannels	"Configuring Layer 3 and Layer 2 EtherChannel" chapter of the *Catalyst 6500 Release 12.2SXF Software Configuration Guide*
Configuring Carrier Ethernet	*Cisco IOS Carrier Ethernet Configuration Guide*
Cisco IOS LACP 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 802.3ad-2000	*IEEE 802.3ad-2000 Link Aggregation*

MIBs

MIB	MIBs Link
802.3ad 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 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 IEEE 802.3ad Link Bundling and Load Balancing

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 21 Feature Information for Configuring IEEE 802.3ad Link Bundling and Load Balancing

Feature Name	Releases	Feature Information
802.3ad Link Aggregation with Weighted Load Balancing	15.0(1)S	The 802.3ad LAG with WLB feature is an enhancement to current load-balancing mechanisms that allows you to assign weights to service instances to efficiently distribute traffic flow across active member links in a port channel. The following commands were introduced or modified: **debug port-channel load-balance**, **port-channel load-balance** (interface), **port-channel load-balance weighted rebalance**, **show ethernet service instance**, **weight**(srvs instance).
EtherChannel Load Distribution	12.2(33)SRC	The EtherChannel Load Distribution feature uses a port reassignment scheme that enhances EtherChannel availability by limiting the load distribution reassignment to the port that is added or deleted. The new load on existing bundled ports does not conflict with the load programmed on those ports when a port is added or deleted. The following commands were introduced or modified: **port-channel port hash-distribution**, **show etherchannel**.
EtherChannel Min-Links	12.2(33)SB 15.0(1)S	The EtherChannel Min-Links feature allows a port channel to be shut down when the number of active links falls below the minimum threshold. Using the **lacp min-bundle** command, you can configure the minimum threshold. The following command was introduced or modified: **lacp min-bundle**.

Feature Name	Releases	Feature Information
IEEE 802.3ad Faster Link Switchover Time	12.2(33)SB	The IEEE 802.3ad Faster Link Switchover Time feature provides a link failover time of 250 milliseconds or less and a maximum link failover time of 2 seconds. Also, port channels remain in the LINK_UP state to eliminate reconvergence by the Spanning-Tree Protocol. The following command was introduced or modified: **lacp fast-switchover**.
IEEE 802.3ad Link Aggregation (LACP)	12.2(31)SB212.2(33)SRB 12.2(33)SRC 15.0(1)S	The IEEE 802.3ad Link Bundling feature provides a method for aggregating multiple Ethernet links into a single logical channel based on the IEEE 802.3ad standard. In addition, this feature provides a capability to dynamically provision, manage, and monitor various aggregated links and enables interoperability between various Cisco devices and devices of third-party vendors. In 12.2(31)SB2, this feature was implemented on the Cisco 10000 series router. In 12.2(33)SRB, this feature was implemented on the Cisco 7600 router. In 12.2(33)SRC, the **lacp rate** command was added. The following commands were introduced or modified: **channel-group (**interface**)**, **debug lacp**, **lacp max-bundle**, **lacp port-priority**, **lacp rate**, **lacp system-priority**, **show lacp**.

Feature Name	Releases	Feature Information
IEEE 802.3ad Maximum Number of Links Increased	12.2(33)SB	The IEEE 802.3ad Maximum Number of Links Increased feature supports eight member links per LACP bundle, an increase from four in previous software releases. This feature uses no new or modified commands.
LACP Single Fault Direct Load Balance Swapping	12.2(33)SRC 15.0(1)S	The LACP Single Fault Direct Load Balance Swapping feature reassigns the load balance bits so that the swapped-in hot standby port is assigned the load balance bits of the failed port, and the load balance bits of the remaining ports in the aggregation remain unchanged. When the swapped-in port is bundled, the load share is recalculated and the stored load share of the failed port is assigned to the swapped-in port. The remaining ports in the bundle are not affected. The following commands were introduced or modified: **lacp direct-loadswap**, **show etherchannel**.
PPPoX Hitless Failover	12.2(33)SB	The PPPoX Hitless Failover feature allows a port channel to remain in the LINK_UP state during a link switchover. In PPPoEoE, PPPoEoQinQ, and PPPoVLAN sessions, both the active and standby links assume the same configured elements after a switchover; the sessions are not forced to reestablish. This feature uses no new or modified commands.

Feature Name	Releases	Feature Information
SSO - LACP	12.2(33)SB	The SSO - LACP feature supports stateful switchover (SSO), In Service Software Upgrade (ISSU), Cisco nonstop forwarding (NSF), and nonstop routing (NSR) on Gigabit EtherChannel bundles. This feature uses no new or modified commands.

Multichassis LACP

In Carrier Ethernet networks, various redundancy mechanisms provide resilient interconnection of nodes and networks. The choice of redundancy mechanisms depends on various factors such as transport technology, topology, single node versus entire network multihoming, capability of devices, autonomous system (AS) boundaries or service provider operations model, and service provider preferences.

Carrier Ethernet network high-availability can be achieved by employing both intra- and interchassis redundancy mechanisms. Cisco's Multichassis EtherChannel (MCEC) solution addresses the need for interchassis redundancy mechanisms, where a carrier wants to "dual home" a device to two upstream points of attachment (PoAs) for redundancy. Some carriers either cannot or will not run loop prevention control protocols in their access networks, making an alternative redundancy scheme necessary. MCEC addresses this issue with enhancements to the 802.3ad Link Aggregation Control Protocol (LACP) implementation. These enhancements are provided in the Multichassis LACP (mLACP) feature described in this document.

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 mLACP

- The command **lacp max-bundle** must be used on all PoAs in order to operate in PoA control and shared control modes.
  - The maximum number of links configured cannot be less than the total number of interfaces in the link aggregation group (LAG) that is connected to the PoA.
  - Each PoA may be connected to a dual-homed device (DHD) with a different number of links for the LAG (configured with a different number of maximum links).
- Each PoA must be configured using the **port-channel min-link**command with the desired minimum number of links to maintain the LAG in the active state.
- Each PoA must be configured with the **errdisable recovery cause mlacp** command if brute-force failover is being used.
- For DHD control there must be an equal number of links going to each PoA.
- The max-bundle value must equal the number of links connected locally to the PoA (no local intra-PoA active or standby protection).
- LACP fast switchover must be configured on all devices to speed convergence.

Restrictions for mLACP

- mLACP does not support Fast Ethernet.
- mLACP does not support half-duplex links.
- mLACP does not support multiple neighbors.
- Converting a port channel to mLACP can cause a service disruption.
- The maximum number of member links per LAG per PoA is restricted by the maximum number of ports per port channel, as limited by the platform.
- System priority on a DHD must be a lesser priority than on PoAs.
- MAC Tunneling Protocol (MTP) supports only one member link in a port channel.
- A port-channel or its member links may flap while LACP stabilizes.
- DHD-based control does not function when min-links is not configured.
- DHD-controlled revertive behavior with min-links is not supported.
- Brute-force failover always causes min-link failures.
- Any failure with brute-force failover behaves revertively.

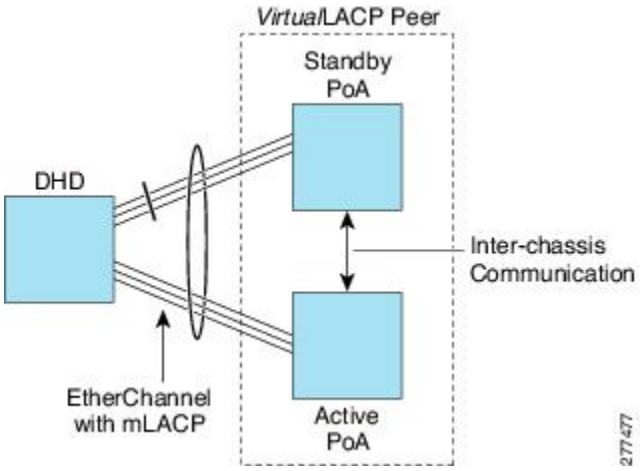
Information About mLACP

Overview of Multichassis EtherChannel

In Multichassis EtherChannel (MCEC), the DHD is dual-homed to two upstream PoAs. The DHD is incapable of running any loop prevention control protocol such as Multiple Spanning Tree (MST).

Therefore, another mechanism is required to prevent forwarding loops over the redundant setup. One method is to place the DHD's uplinks in a LAG, commonly referred to as EtherChannel. This method assumes that the DHD is capable of running only IEEE 802.3ad LACP for establishing and maintaining the LAG.

LACP, as defined in IEEE 802.3ad, is a link-level control protocol that allows the dynamic negotiation and establishment of LAGs. An extension of the LACP implementation to PoAs is required to convey to a DHD that it is connected to a single virtual LACP peer and not to two disjointed devices. This extension is called Multichassis LACP or mLACP. The figure below shows this setup.



The PoAs forming a virtual LACP peer, from the perspective of the DHD, are defined as members of a redundancy group. For the PoAs in a redundancy group to appear as a single device to the DHD, the states between them must be synchronized through the Interchassis Communication Protocol (ICCP), which provides a control-only interchassis communication channel (ICC).

In Cisco IOS Release 12.2(33)SRE, the system functions in active/standby redundancy mode. In this mode DHD uplinks that connect to only a single PoA can be active at any time. The DHD recognizes one PoA as active and the other as standby but does not preclude a given PoA from being active for one DHD and standby for another. This capability allows two PoAs to perform load sharing for different services.

Interactions with the MPLS Pseudowire Redundancy Mechanism

The network setup shown in the figure above can be used to provide provider edge (PE) node redundancy for Virtual Private LAN Service (VPLS) and Virtual Private Wire Service (VPWS) deployments over Multiprotocol Label Switching (MPLS). In these deployments, the uplinks of the PoAs host the MPLS pseudowires that provide redundant connectivity over the core to remote PE nodes. Proper operation of the network requires interaction between the redundancy mechanisms employed on the attachment circuits (for example, mLACP) and those employed on the MPLS pseudowires. This interaction ensures the state (active or standby) is synchronized between the attachment circuits and pseudowires for a given PoA.

RFC 4447 introduced a mechanism to signal pseudowire status via the Link Distribution Protocol (LDP) and defined a set of status codes to report attachment circuit as well as pseudowire fault information. The Preferential Forwarding Status bit (*draft-ietf-pwe3-redundancy-bit*) definition proposes to extend these codes to include two bits for pseudowire redundancy applications:

- Preferential forwarding status: active or standby
- Request pseudowire switchover

The draft also proposes two modes of operation:

- Independent mode--The local PE decides on its pseudowire status independent of the remote PE.
- Primary and secondary modes--One of the PEs determines the state of the remote side through a handshake mechanism.

For the mLACP feature, operation is based on the independent mode. By running ICC between the PoAs, only the preferential forwarding status bit is required; the request pseudowire switchover bit is not used.

The local pseudowire status (active or standby) is determined independently by the PoAs in a redundancy group and then relayed to the remote PEs in the form of a notification. Similarly, the remote PEs perform their own selection of their pseudowire status and notify the PoAs on the other side of the core.

After this exchange of local states, the pseudowires used for traffic forwarding are those selected to be active independently on both local and remote ends.

The attachment circuit redundancy mechanism determines and controls the pseudowire redundancy mechanism. mLACP determines the status of the attachment circuit on a given PoA according to the configured LACP system and port priorities, and then the status of the pseudowires on a given PoA is synchronized with that of the local attachment circuits. This synchronization guarantees that the PoA with the active attachment circuits has its pseudowires active. Similarly, the PoA with the standby attachment circuits has its pseudowires in standby mode. By ensuring that the forwarding status of the attachment circuits is synchronized with that of the pseudowires, the need to forward data between PoA nodes within a redundancy group can be avoided. This synchronization saves platform bandwidth that would otherwise be wasted on inter-PoA data forwarding in case of failures.

Redundancy Mechanism Processes

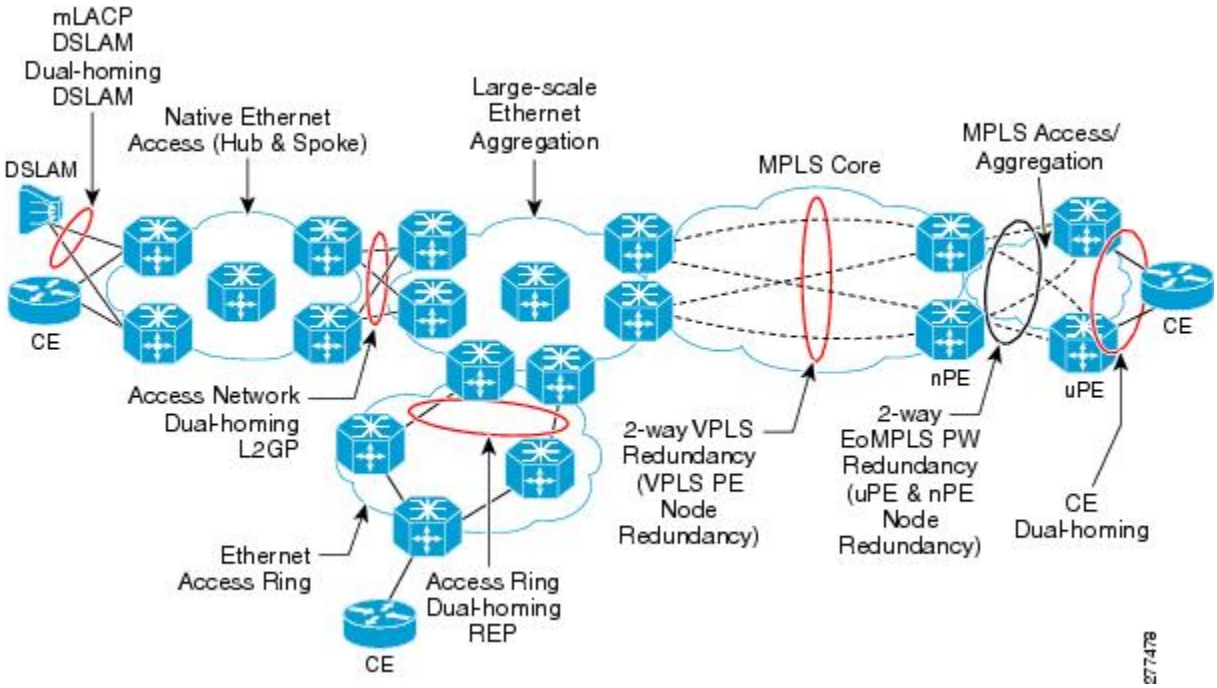
The Carrier Ethernet redundancy solution should include the following processes (and how they apply to the mLACP solution):

- Attachment circuit active or standby status selection--This selection can be performed by the access node or network, the aggregation node, or combination of the two. For mLACP, the attachment circuit status selection is determined through collaboration between the DHD and the PoAs.
- Pseudowire forwarding status notification--This notification is mandatory for mLACP operation in VPWS and VPLS deployments; that is, when the PoA uplinks employ pseudowire technology. When the PoAs decide on either an active or standby role, they need to signal the status of the associated pseudowires to the PEs on the far end of the network. For MPLS pseudowires, this is done using LDP.
- MAC flushing indication--This indication is mandatory for any redundancy mechanism in order to speed convergence time and eliminate potential traffic blackholing. The mLACP redundancy mechanism should be integrated with relevant 802.1Q/802.1ad/802.1ah MAC flushing mechanisms as well as MAC flushing mechanisms for VPLS.

Note Blackholing occurs when incoming traffic is dropped without informing the source that the data did not reach its intended recipient. A black hole can be detected only when lost traffic is monitored.

- Active VLAN notification--For mLACP, this notification is not required as long as the PoAs follow the active/standby redundancy model.

The figure below shows redundancy mechanisms in Carrier Ethernet networks.



Dual-Homed Topology Using mLACP

The mLACP feature allows the LACP state machine and protocol to operate in a dual-homed topology. The mLACP feature decouples the existing LACP implementation from the multichassis specific requirements, allowing LACP to maintain its adherence to the IEEE 802.3ad standard. The mLACP feature exposes a single virtual instance of IEEE 802.3ad to the DHD for each redundancy group. The virtual LACP instance interoperates with the DHD according to the IEEE 802.3ad standard to form LAGs spanning two or more chassis.

LACP and 802.3ad Parameter Exchange

In IEEE 802.3ad, the concatenation of the LACP system MAC address and system priority form an LACP system ID (8 bytes). The system ID is formed by taking the two-byte system priority value as the most significant two octets of the system ID. The system MAC address makes up the remainder of the system ID (octets 3 to 8). System ID priority comparisons are based on the lower numerically valued ID.

To provide the highest LACP priority, the mLACP module communicates the system MAC address and priority values for the given redundancy group to its redundancy group peer(s) and vice versa. The mLACP then chooses the lowest system ID value among the PoAs in the given redundancy group to use as the system ID of the virtual LACP instance of the redundancy group.

Cisco IOS Release 12.2(33)SRE introduces two LACP configuration commands to specify the system MAC address and system priority used for a given redundancy group: **mlacp system-mac** *mac-address* and **mlacp system-priority** *priority-value*. These commands provide better settings to determine which side of the attachment circuit will control the selection logic of the LAG. The default value for the system MAC address is the chassis backplane default MAC address. The default value for the priority is 32768.

Port Identifier

IEEE 802.3ad uses a 4-byte port identifier to uniquely identify a port within a system. The port identifier is the concatenation of the port priority and port number (unique per system) and identifies each port in the system. Numerical comparisons between port IDs are performed by unsigned integer comparisons where the 2-byte Port Priority field is placed in the most significant two octets of the port ID. The 2-byte port number makes up the third and fourth octets. The mLACP feature coordinates the port IDs for a given redundancy group to ensure uniqueness.

Port Number

A port number serves as a unique identifier for a port within a device. The LACP port number for a port is equal to the port's ifIndex value (or is based on the slot and subslot identifiers on the Cisco 7600 router).

LACP relies on port numbers to detect rewiring. For multichassis operation, you must enter the **mlacp node-id** *node-id* command to coordinate port numbers between the two PoAs in order to prevent overlap.

Port Priority

Port priority is used by the LACP selection logic to determine which ports should be activated and which should be left in standby mode when there are hardware or software limitations on the maximum number of links allowed in a LAG. For multichassis operation in active/standby redundancy mode, the port priorities for all links connecting to the active PoA must be higher than the port priorities for links connecting to the standby PoA. These port priorities can either be guaranteed through explicit configuration or the system can automatically adjust the port priorities depending on selection criteria. For example, select the PoA with the highest port priority to be the active PoA and dynamically adjust the priorities of all other links with the same port key to an equal value.

In Cisco IOS Release 12.2(33)SRE, the mLACP feature supports only the active/standby redundancy model. The LACP port priorities of the individual member links should be the same for each link belonging to the LAG of a given PoA. To support this requirement, the **mlacp lag-priority** command is implemented in interface configuration mode in the command-line interface (CLI). This command sets the LACP port priorities for all the local member links in the LAG. Individual member link LACP priorities (configured by the **lacp port-priority** command) are ignored on links belonging to mLACP port channels.

The **mlacp lag-priority** command may also be used to force a PoA failover during operation in the following two ways:

- Set the active PoA's LAG priority to a value greater than the LAG priority on the standby PoA. This setting results in the quickest failover because it requires the fewest LACP link state transitions on the standby links before they turn active.
- Set the standby PoA's LAG priority to a value numerically less than the LAG priority on the active PoA. This setting results in a slightly longer failover time because standby links have to signal OUT_OF_SYNC to the DHD before the links can be brought up and go active.

In some cases, the operational priority and the configured priority may differ when using dynamic port priority management to force failovers. In this case, the configured version will not be changed unless the port channel is operating in nonrevertive mode. Enter the **show lacp multichassis port-channel** command

to view the current operational priorities. The configured priority values can be displayed by using the **show running-config** command.

Multichassis Considerations

Because LACP is a link layer protocol, all messages exchanged over a link contain information that is specific and local to that link. The exchanged information includes:

- System attributes--priority and MAC address
- Link attributes--port key, priority, port number, and state

When extending LACP to operate over a multichassis setup, synchronization of the protocol attributes and states between the two chassis is required.

System MAC Address

LACP relies on the system MAC address to determine the identity of the remote device connected over a particular link. Therefore, to mask the DHD from its connection to two disjointed devices, coordination of the system MAC address between the two PoAs is essential. In Cisco IOS software, the LACP system MAC address defaults to the ROM backplane base MAC address and cannot be changed by configuration. For multichassis operation the following two conditions are required:

- System MAC address for each PoA should be communicated to its peer--For example, the PoAs elect the MAC address with the lower numeric value to be the system MAC address. The arbitration scheme must resolve to the same value. Choosing the lower numeric MAC address has the advantage of providing higher system priority.
- System MAC address is configurable--The system priority depends, in part, on the MAC address, and a service provider would want to guarantee that the PoAs have higher priority than the DHD (for example, if both DHD and PoA are configured with the same system priority and the service provider has no control over DHD). A higher priority guarantees that the PoA port priorities take precedence over the DHD’s port priority configuration. If you configure the system MAC address, you must ensure that the addresses are uniform on both PoAs; otherwise, the system will automatically arbitrate the discrepancy, as when a default MAC address is selected.

System Priority

LACP requires that a system priority be associated with every device to determine which peer’s port priorities should be used by the selection logic when establishing a LAG. In Cisco IOS software, this parameter is configurable through the CLI. For multichassis operation, this parameter is coordinated by the PoAs so that the same value is advertised to the DHD.

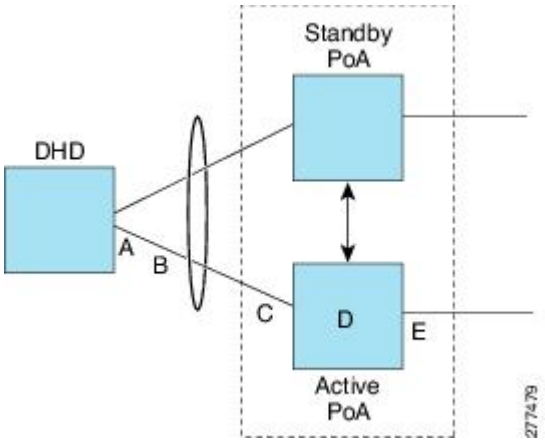
Port Key

The port key indicates which links can form a LAG on a given system. The key is locally significant to an LACP system and need not match the key on an LACP peer. Two links are candidates to join the same LAG if they have the same key on the DHD and the same key on the PoAs; however, the key on the DHD is not required to be the same as the key on the PoAs. Given that the key is configured according to the need to aggregate ports, there are no special considerations for this parameter for multichassis operation.

Failure Protection Scenarios

The mLACP feature provides network resiliency by protecting against port, link, and node failures. These failures can be categorized into five types. The figure below shows the failure points in a network, denoted by the letters A through E.

- A--Failure of the uplink port on the DHD
- B--Failure of the Ethernet link
- C--Failure of the downlink port on the active PoA
- D--Failure of the active PoA node
- E--Failure of the active PoA uplinks



When any of these faults occur, the system reacts by triggering a switchover from the active PoA to the standby PoA. The switchover involves failing over the PoA's uplinks and downlinks simultaneously.

Failure points A and C are port failures. Failure point B is an Ethernet link failure and failure point D is a node failure. Failure point E can represent one of four different types of uplink failures when the PoAs connect to an MPLS network:

- Pseudowire failure--Monitoring individual pseudowires (for example, using VCCV-BFD) and, upon a pseudowire failure, declare uplink failure for the associated service instances.
- Remote PE IP path failure--Monitoring the IP reachability to the remote PE (for example, using IP Route-Watch) and, upon route failure, declare uplink failure for all associated service instances.
- LSP failure--Monitoring the LSP to a given remote PE (for example, using automated LSP-Ping) and, upon LSP failure, declare uplink failure for all associated service instances.
- PE isolation--Monitoring the physical core-facing interfaces of the PE. When all of these interfaces go down, the PE effectively becomes isolated from the core network, and the uplink failure is declared for all affected service instances.

As long as the IP/MPLS network employs native redundancy and resiliency mechanisms such as MPLS fast reroute (FRR), the mLACP solution is sufficient for providing protection against PE isolation. Pseudowire, LSP, and IP path failures are managed by the native IP/MPLS protection procedures. That is, interchassis failover via mLACP is triggered only when a PE is completely isolated from the core network, because native IP/MPLS protection mechanisms are rendered useless. Therefore, failure point E is used to denote PE isolation from the core network.

Note The set of core-facing interfaces that should be monitored are identified by explicit configuration. The set of core-facing interfaces must be defined independently per redundancy group. Failure point E (unlike failure point A, B, or C) affects and triggers failover for all the multichassis LAGs configured on a given PoA.

Operational Variants

LACP provides a mechanism by which a set of one or more links within a LAG are placed in standby mode to provide link redundancy between the devices. This redundancy is normally achieved by configuring more ports with the same key than the number of links a device can aggregate in a given LAG (due to hardware or software restrictions, or due to configuration). For active/standby redundancy, two ports are configured with the same port key, and the maximum number of allowed links in a LAG is configured to be 1. If the DHD and PoAs are all capable of restricting the number of links per LAG by configuration, three operational variants are possible.

- DHD-based Control, page 471
- PoA Control, page 472
- Shared Control (PoA and DHD), page 472

DHD-based Control

The DHD is configured to limit the maximum number of links per bundle to one, whereas the PoAs are configured to limit the maximum number of links per bundle to greater than one. Thus, the selection of the active/standby link is the responsibility of the DHD. Which link is designated active and which is marked standby depends on the relative port priority, as configured on the system with the higher system priority. A PoA configured with a higher system priority can still determine the selection outcome. The DHD makes the selection and places the link with lower port priority in standby mode.

To accommodate DHD-controlled failover, the DHD must be configured with the max-bundle value equal to a number of links (L), where L is the fewest number of links connecting the DHD to a PoA. The max-bundle value restricts the DHD from bundling links to both PoAs at the same time (active/active). Although the DHD controls the selection of active/standby links, the PoA can still dictate the individual member link priorities by configuring the PoA's virtual LACP instance with a lower system priority value than the DHD's system priority.

The DHD control variant must be used with a PoA minimum link threshold failure policy where the threshold is set to L (same value for L as described above). A minimum link threshold must be configured on each of the PoAs because an A, B, or C link failure that does not trigger a failover (minimum link threshold is still satisfied) causes the DHD to add one of the standby links going to the standby PoA to the bundle. This added link results in the unsupported active/active scenario.

Note DHD control does not use the mLACP hot-standby state on the standby PoA, which results in higher failover times than the other variants.

DHD control eliminates the split brain problem on the attachment circuit side by limiting the DHD's attempts to bundle all the links.

PoA Control

In PoA control, the PoA is configured to limit the maximum number of links per bundle to be equal to the number of links (L) going to the PoA. The DHD is configured with that parameter set to some value greater than L. Thus, the selection of the active/standby links becomes the responsibility of the PoA.

Shared Control (PoA and DHD)

In shared control, both the DHD and the PoA are configured to limit the maximum number of links per bundle to L--the number of links going to the PoA. In this configuration, each device independently selects the active/standby link. Shared control is advantageous in that it limits the split-brain problem in the same manner as DHD control, and shared control is not susceptible to the active/active tendencies that are prevalent in DHD control. A disadvantage of shared control is that the failover time is determined by both the DHD and the PoA, each changing the standby links to SELECTED and waiting for each of the WAIT_WHILE_TIMERs to expire before moving the links to IN_SYNC. The independent determination of failover time and change of link states means that both the DHD and PoAs need to support the LACP fast-switchover feature in order to provide a failover time of less than one second.

mLACP Failover

The mLACP forces a PoA failover to the standby PoA when one of the following failures occurs:

- Failure of the DHD uplink port, Ethernet link, or downlink port on the active PoA--A policy failover is triggered via a configured failover policy and is considered a forced failover. In Cisco IOS Release 12.2(33)SRE, the only option is the configured minimum bundle threshold. When the number of active and SELECTED links to the active PoA goes below the configured minimum threshold, mLACP forces a failover to the standby PoA's member links. This minimum threshold is configured using the **port-channel min-links** command in interface configuration mode. The PoAs determine the failover independent of the operational control variant in use.
- Failure of the active PoA--This failure is detected by the standby PoA. mLACP automatically fails over to standby because mLACP on the standby PoA is notified of failure via ICRM and brings up its local member links. In the DHD-controlled variant, this failure looks the same as a total member link failure, and the DHD activates the standby links.
- Failure of the active PoA uplinks--mLACP is notified by ICRM of PE isolation and relinquishes its active member links. This failure is a "forced failover" and is determined by the PoAs independent of the operational control variant in use.

- Dynamic Port Priority, page 472
- Revertive and Nonrevertive Modes, page 473
- Brute Force Shutdown, page 473
- Peer Monitoring with Interchassis Redundancy Manager, page 473
- MAC Flushing Mechanisms, page 475

Dynamic Port Priority

The default failover mechanism uses dynamic port priority changes on the local member links to force the LACP selection logic to move the required standby link(s) to the SELECTED and Collecting_Distributing state. This state change occurs when the LACP actor port priority values for all affected member links on the currently active PoA are changed to a higher numeric value than the standby PoA's port priority (which gives the standby PoA ports a higher claim to bundle links). Changing the actor port priority triggers the

transmission of an mLACP Port Config Type-Length-Value (TLV) message to all peers in the redundancy group. These messages also serve as notification to the standby PoA(s) that the currently active PoA is attempting to relinquish its role. The LACP then transitions the standby link(s) to the SELECTED state and moves all the currently active links to STANDBY.

Dynamic port priority changes are not automatically written back to the running configuration or to the NVRAM configuration. If you want the current priorities to be used when the system reloads, the **mlacp lag-priority** command must be used and the configuration must be saved.

Revertive and Nonrevertive Modes

Dynamic port priority functionality is used by the mLACP feature to provide both revertive mode and nonrevertive mode. The default operation is revertive, which is the default behavior in single chassis LACP. Nonrevertive mode can be enabled on a per port-channel basis by using the **lacp failover non-revertive**command in interface configuration mode. In Cisco IOS Release 12.2(33)SRE this command is supported only for mLACP.

Nonrevertive mode is used to limit failover and, therefore, possible traffic loss. Dynamic port priority changes are utilized to ensure that the newly activated PoA remains active after the failed PoA recovers.

Revertive mode operation forces the configured primary PoA to return to active state after it recovers from a failure. Dynamic port priority changes are utilized when necessary to allow the recovering PoA to resume its active role.

Brute Force Shutdown

A brute-force shutdown is a forced failover mechanism to bring down the active physical member link interface(s) for the given LAG on the PoA that is surrendering its active status. This mechanism does not depend on the DHD's ability to manage dynamic port priority changes and compensates for deficiencies in the DHD's LACP implementation.

The brute-force shutdown changes the status of each member link to ADMIN_DOWN to force the transition of the standby links to the active state. Note that this process eliminates the ability of the local LACP implementation to monitor the link state.

The brute-force shutdown operates in revertive mode, so dynamic port priorities cannot be used to control active selection. The brute-force approach is configured by the **lacp failover brute-force** command in interface configuration mode. This command is not allowed in conjunction with a nonrevertive configuration.

Peer Monitoring with Interchassis Redundancy Manager

There are two ways in which a peer can be monitored with Interchassis Redundancy Manager (ICRM):

- Routewatch (RW)--This method is the default.
- Bidirectional Forwarding Detection (BFD)--You must configure the redundancy group with the **monitor peer bfd** command.

Note For stateful switchover (SSO) deployments (with redundant support in the chassis), BFD monitoring and a static route for the ICCP connection are required to prevent "split brain" after an SSO failover. Routewatch is compatible with SSO for health monitoring.

For each redundancy group, for each peer (member IP), a monitoring adjacency is created. If there are two peers with the same IP address, the adjacency is shared regardless of the monitoring mode. For example, if

redundancy groups 1 and 2 are peered with member IP 10.10.10.10, there is only one adjacency to 10.10.10.10, which is shared in both redundancy groups. Furthermore, redundancy group 1 can use BFD monitoring while redundancy group 2 is using RW.

Note BFD is completely dependent on RW--there must be a route to the peer for ICRM to initiate BFD monitoring. BFD implies RW and sometimes the status of the adjacency may seem misleading but is accurately representing the state. Also, if the route to the peer PoA is not through the directly connected (back-to-back) link between the systems, BFD can give misleading results.

An example of output from the **show redundancy interface** command follows:

```
Router# show redundancy interface
Redundancy Group 1 (0x1)
  Applications connected: mLACP
  Monitor mode: Route-watch
  member ip: 201.0.0.1 'mlacp-201', CONNECTED
    Route-watch for 201.0.0.1 is UP
    mLACP state: CONNECTED
ICRM fast-failure detection neighbor table
  IP Address       Status Type Next-hop IP      Interface
  ==========       ====== ==== ===========      =========
  201.0.0.1        UP     RW
```

To interpret the adjacency status displayed by the **show redundancy interchassis**command, refer to the table below.

Table 22 Status Information from the show redundancy interchassis command

Adjacency Type	Adjacency Status	Meaning
RW	DOWN	RW or BFD is configured, but there is no route for the given IP address.
RW	UP	RW or BFD is configured. RW is up, meaning there is a valid route to the peer. If BFD is configured and the adjacency status is UP, BFD is probably not configured on the interface of the route's adjacency.
BFD	DOWN	BFD is configured. A route exists and the route's adjacency is to an interface that has BFD enabled. BFD is started but the peer is down. The DOWN status can be because the peer is not present or BFD is not configured on the peer's interface.
BFD	UP	BFD is configured and operational.

Note If the adjacency type is "BFD," RW is UP regardless of the BFD status.

MAC Flushing Mechanisms

When mLACP is used to provide multichassis redundancy in multipoint bridged services (for example, VPLS), there must be a MAC flushing notification mechanism in order to prevent potential traffic blackholing.

At the failover from a primary PoA to a secondary PoA, a service experiences traffic blackholing when the DHD in question remains inactive and while other remote devices in the network are attempting to send traffic to that DHD. Remote bridges in the network have stale MAC entries pointing to the failed PoA and direct traffic destined to the DHD to the failed PoA, where the traffic is dropped. This blackholing continues until the remote devices age out their stale MAC address table entries (which typically takes five minutes). To prevent this anomaly, the newly active PoA, which has taken control of the service, transmits a MAC flush notification message to the remote devices in the network to flush their stale MAC address entries for the service in question.

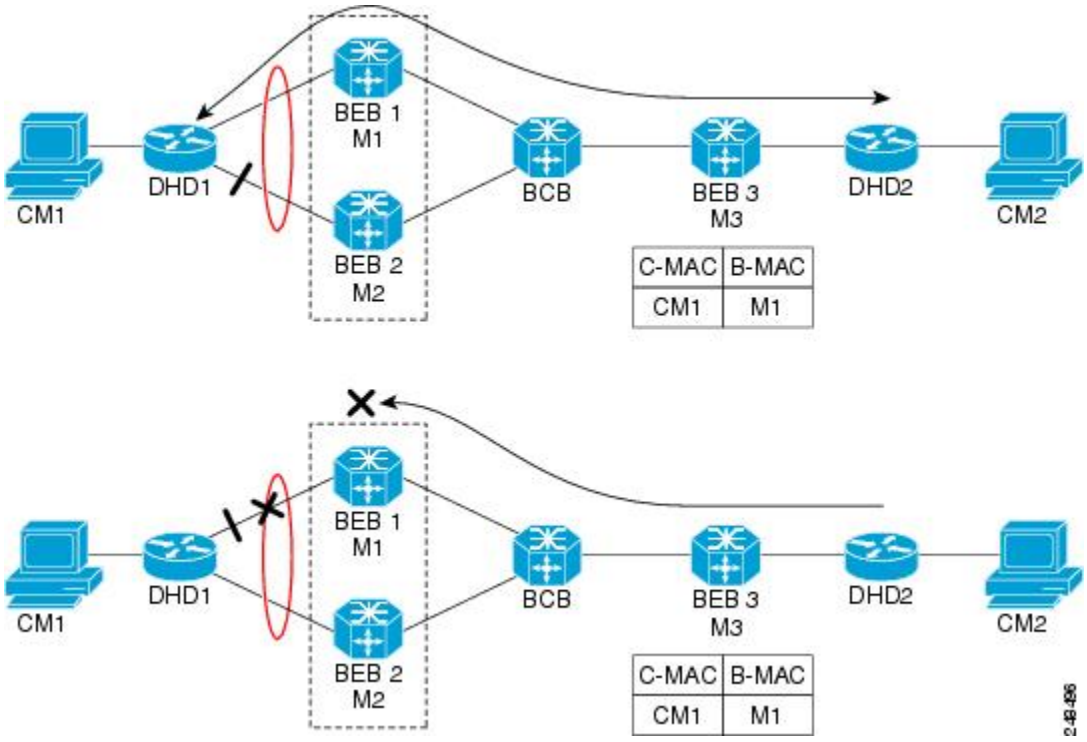
The exact format of the MAC flushing message depends on the nature of the network transport: native 802.1Q/802.1ad Ethernet, native 802.1ah Ethernet, VPLS, or provider backbone bridge (PBB) over VPLS. Furthermore, in the context of 802.1ah, it is important to recognize the difference between mechanisms used for customer-MAC (C-MAC) address flushing versus bridge-MAC (B-MAC) address flushing.

The details of the various mechanisms are discussed in the following sections.

- Multiple I-SID Registration Protocol, page 475
- LDP MAC Address Withdraw, page 477

Multiple I-SID Registration Protocol

Multiple I-SID Registration Protocol (MIRP) is enabled by default on 802.1ah service instances. The use of MIRP in 802.1ah networks is shown in the figure below.



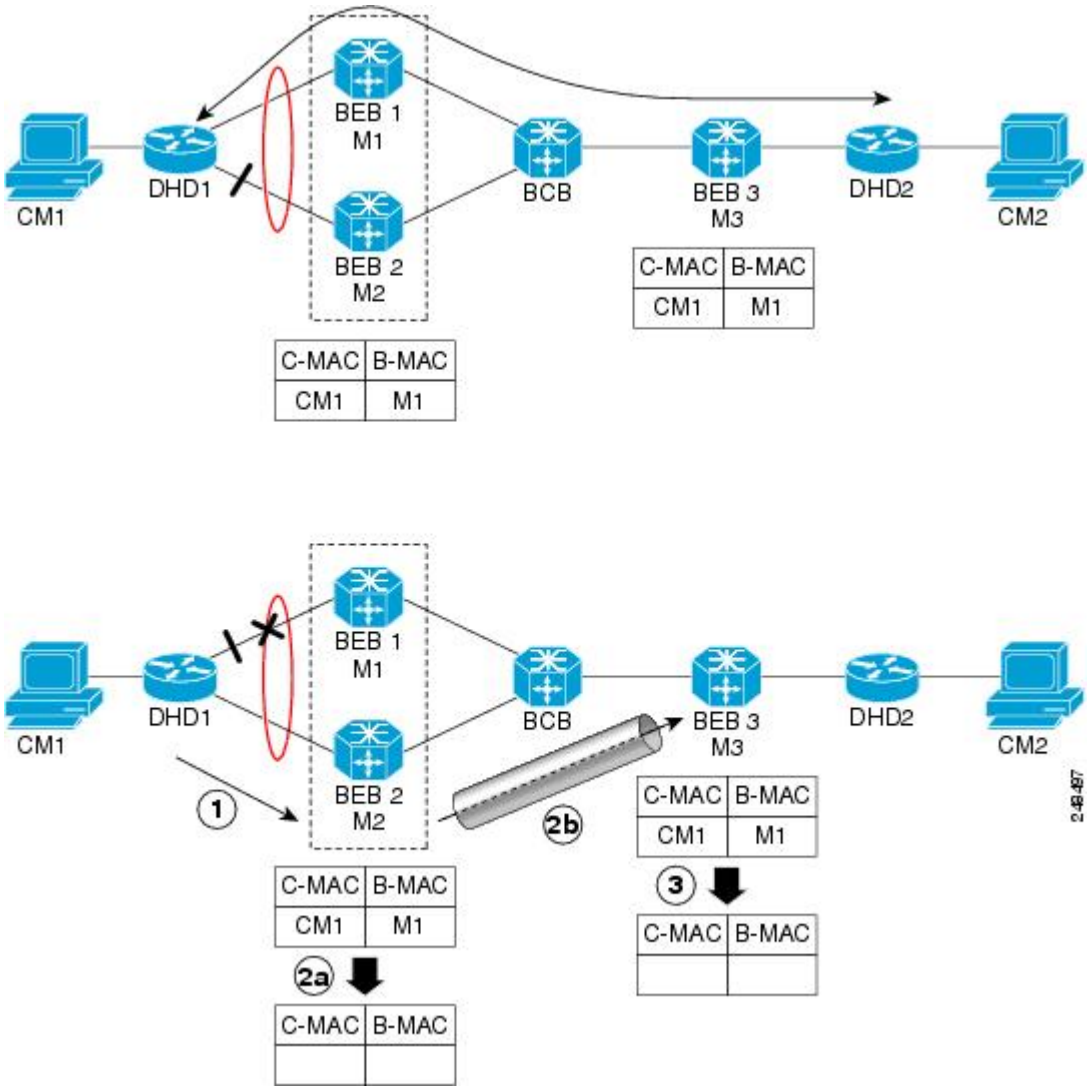
Device DHD1 is dual-homed to two 802.1ah backbone edge bridges (BEB1 and BEB2). Assume that initially the primary path is through BEB1. In this configuration BEB3 learns that the host behind DHD1 (with MAC address CM1) is reachable via the destination B-MAC M1. If the link between DHD1 and BEB1 fails and the host behind DHD1 remains inactive, the MAC cache tables on BEB3 still refer to the BEB1 MAC address even though the new path is now via BEB2 with B-MAC address M2. Any bridged traffic destined from the host behind DHD2 to the host behind DHD1 is wrongfully encapsulated with B-MAC M1 and sent over the MAC tunnel to BEB1, where the traffic blackholes.

To circumvent the traffic blackholing problem when the link between DHD1 and BEB1 fails, BEB2 performs two tasks:

- Flushes its own MAC address table for the service or services in question.
- Transmits an MIRP message on its uplink to signal the far end BEB (BEB3) to flush its MAC address table. Note that the MIRP message is transparent to the backbone core bridges (BCBs). The MIRP message is processed on a BEB because only BCBs learn and forward based on B-MAC addresses and they are transparent to C-MAC addresses.

Note MIRP triggers C-MAC address flushing for both native 802.1ah and PBB over VPLS.

The figure below shows the operation of the MIRP.



The MIRP has not been defined in IEEE but is expected to be based on the IEEE 802.1ak Multiple Registration Protocol (MRP). MRP maintains a complex finite state machine (FSM) for generic attribute registration. In the case of MIRP, the attribute is an I-SID. As such, MIRP provides a mechanism for BEBs to build and prune a per I-SID multicast tree. The C-MAC flushing notification capability of MIRP is a special case of attribute registration in which the device indicates that an MIRP declaration is “new,” meaning that this notification is the first time a BEB is declaring interest in a particular I-SID.

LDP MAC Address Withdraw

When the mLACP feature is used for PE redundancy in traditional VPLS (that is, not PBB over VPLS), the MAC flushing mechanism is based on the LDP MAC Address Withdraw message as defined in RFC 4762.

The required functional behavior is as follows: Upon a failover from the primary PoA to the standby PoA, the standby PoA flushes its local MAC address table for the affected services and generates the LDP MAC Address Withdraw messages to notify the remote PEs to flush their own MAC address tables. One message is generated for each pseudowire in the affected virtual forwarding instances (VFIs).

How to Configure mLACP

Configuring Interchassis Group and Basic mLACP Commands

Perform this task to set up the communication between multiple PoAs and to configure them in the same group.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **redundancy**
4. **interchassis group** *group-id*
5. **monitor peer bfd**
6. **member ip** *ip-address*
7. **mlacp node-id node-id**
8. **mlacp system-mac mac-address**
9. **mlacp system-priority** *priority-value*
10. **backbone interface** *type number*
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.

	Command or Action	Purpose
Step 3	**redundancy** **Example:** `Router(config)# redundancy`	Enters redundancy configuration mode.
Step 4	**interchassis group** *group-id* **Example:** `Router(config-red)# interchassis group 50`	Configures an interchassis group within the redundancy configuration mode and enters interchassis redundancy mode.
Step 5	**monitor peer bfd** **Example:** `Router(config-r-ic)# monitor peer bfd`	Configures the BFD option to monitor the state of the peer. The default option is route-watch.
Step 6	**member ip** *ip-address* **Example:** `Router(config-r-ic)# member ip 172.3.3.3`	Configures the IP address of the mLACP peer member group.
Step 7	**mlacp node-id node-id** **Example:** `Router(config-r-ic)# mlacp node-id 5`	Defines the node ID used in the LACP Port ID field by this member of the mLACP redundancy group. • The valid range is 0 to 7, and the value should be different from the peer values.
Step 8	**mlacp system-mac mac-address** **Example:** `Router(config-r-ic)# mlacp system-mac aa12.be45.d799`	Defines and advertises the system MAC address value to the mLACP members of the redundancy group for arbitration. • The format of the *mac-address* argument must be in standard MAC address format: aabb.ccdd.eeff.
Step 9	**mlacp system-priority** *priority-value* **Example:** `Router(config-r-ic)# mlacp system-priority 100`	Defines the system priority advertised to the other mLACP members of the redundancy group. • System priority values are 1 to 65535. Default value is 32768. • The assigned values should be lower than the DHD.

	Command or Action	Purpose
Step 10	**backbone interface** *type number* **Example:** `Router(config-r-ic)# backbone interface GigabitEthernet2/3`	Defines the backbone interface for the mLACP configuration.
Step 11	**end** **Example:** `Router(config-r-ic)# end`	Returns the CLI to privileged EXEC mode.

Configuring the mLACP Interchassis Group and Other Port-Channel Commands

Perform this task to set up mLACP attributes specific to a port channel. The **mlacp interchassis group** command links the port-channel interface to the interchassis group that was created in the previous Configuring Interchassis Group and Basic mLACP Commands, page 478.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface port-channel** *port-channel- number*
4. **lacp max-bundle** *max-bundles*
5. **lacp failover** {**brute-force**| **non-revertive**}
6. **exit**
7. **redundancy**
8. **interchassis group** *group-id*
9. **exit**
10. **exit**
11. **errdisable recovery cause mlacp-minlink**
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	**interface port-channel** *port-channel- number* **Example:** `Router(config)# interface port-channel1`	Configures the port channel and enters interface configuration mode.
Step 4	**lacp max-bundle** *max-bundles* **Example:** `Router(config-if)# lacp max-bundle 4`	Configures the max-bundle links that are connected to the PoA. The value of the *max-bundles* argument should not be less than the total number of links in the LAG that are connected to the PoA. • Determines whether the redundancy group is under DHD control, PoA control, or both. • Range is 1 to 8. Default value is 8.
Step 5	**lacp failover** {**brute-force**| **non-revertive**} **Example:** `Router(config-if)# lacp failover brute-force`	Sets the mLACP switchover to nonrevertive or brute force. This command is optional. • Default value is revertive (with 180-second delay). • If you configure brute force, a minimum link failure for every mLACP failure occurs or the dynamic lag priority value is modified.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 7	**redundancy** **Example:** `Router(config)# redundancy`	Enters redundancy configuration mode.

	Command or Action	Purpose
Step 8	**interchassis group** *group-id* **Example:** `Router(config-red)# interchassis group 230`	Specifies that the port channel is an mLACP port channel. The *group-id* should match the configured redundancy group.
Step 9	**exit** **Example:** `Router(config-r-ic)# exit`	Exits interchassis redundancy mode.
Step 10	**exit** **Example:** `Router(config-red)# exit`	Exits redundancy configuration mode.
Step 11	**errdisable recovery cause mlacp-minlink** **Example:** `Router(config)# errdisable recovery cause mlacp-minlink`	Enables automatic recovery from a failover state of the port channel.
Step 12	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Configuring Redundancy for VPWS

Perform this task to provide Layer 2 VPN service redundancy for VPWS.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *pw-class-name*
4. **encapsulation mpls**
5. **status peer topology dual-homed**
6. **exit**
7. **interface port-channel** *port-channel-number*
8. **no ip address**
9. **lacp fast-switchover**
10. **lacp max-bundle** *max-bundles*
11. **exit**
12. **redundancy**
13. **interchassis group** *group-id*
14. **exit**
15. **exit**
16. **interface port-channel** *port-channel-number*
17. **service instance** *id* **ethernet** [*evc-name*]
18. **encapsulation dot1q** *vlan-id* [**,** *vlan-id*[**-** *vlan-id*]] [**native**]
19. **exit**
20. **xconnect** *peer-ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} [**pw-class** *pw-class-name*] [**sequencing** {**transmit** | **receive** | **both**}]
21. **backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*] [**priority** *value*]
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	**pseudowire-class** *pw-class-name* **Example:** `Router(config)# pseudowire-class ether-pw`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies that MPLS is used as the data encapsulation method for tunneling Layer 2 traffic over the pseudowire.
Step 5	**status peer topology dual-homed** **Example:** `Router(config-pw-class)# status peer topology dual-homed`	Enables the reflection of the attachment circuit status onto both the primary and secondary pseudowires. This condition is necessary if the peer PEs are connected to a dual-homed device.
Step 6	**exit** **Example:** `Router(config-pw-class)# exit`	Exits pseudowire class configuration mode.
Step 7	**interface port-channel** *port-channel-number* **Example:** `Router(config)# interface port-channel1`	Configures the port channel and enters interface configuration mode.
Step 8	**no ip address** **Example:** `Router(config-if)# no ip address`	Specifies that the VLAN interface does not have an IP address assigned to it.
Step 9	**lacp fast-switchover** **Example:** `Router(config-if)# lacp fast-switchover`	Enables LACP 1-to-1 link redundancy.

	Command or Action	Purpose
Step 10	**lacp max-bundle** *max-bundles* **Example:** `Router(config-if)# lacp max-bundle 4`	Configures the max-bundle links that are connected to the PoA. The value of the *max-bundles*argument should not be less than the total number of links in the LAG that are connected to the PoA. • Determines whether the redundancy group is under DHD control, PoA control, or both. • Range is 1 to 8. Default value is 8.
Step 11	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 12	**redundancy** **Example:** `Router(config)# redundancy`	Enters redundancy configuration mode.
Step 13	**interchassis group** *group-id* **Example:** `Router(config-red)# interchassis group 230`	Specifies that the port channel is an mLACP port channel. • The *group-id* should match the configured redundancy group.
Step 14	**exit** **Example:** `Router(config-r-ic)# exit`	Exits interchassis redundancy mode.
Step 15	**exit** **Example:** `Router(config-red)# exit`	Exits redundancy configuration mode.
Step 16	**interface port-channel** *port-channel-number* **Example:** `Router(config)# interface port-channel1`	Configures the port channel and enters interface configuration mode.

	Command or Action	Purpose
Step 17	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Router(config-if)# service instance 1 ethernet`	Configures an Ethernet service instance.
Step 18	**encapsulation dot1q** *vlan-id* [**,** *vlan-id*[**-** *vlan-id*]] [**native**] **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
Step 19	**exit** **Example:** `Router(config-if-srv)# exit`	Exits service instance configuration mode.
Step 20	**xconnect** *peer-ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} [**pw-class** *pw-class-name*] [**sequencing** {**transmit** | **receive** | **both**}] **Example:** `Router(config-if)# xconnect 10.0.3.201 123 pw-class ether-pw`	Binds an attachment circuit to a pseudowire.
Step 21	**backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*] [**priority** *value*] **Example:** `Router(config-if)# backup peer 10.1.1.1 123 pw-class ether-pw`	Specifies a redundant peer for a pseudowire virtual circuit.
Step 22	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Configuring Redundancy for VPLS

Coupled and Decoupled Modes for VPLS

VPLS can be configured in either coupled mode or decoupled mode. Coupled mode is when at least one attachment circuit in VFI changes state to active, all pseudowires in VFI advertise active. When all attachment circuits in VFI change state to standby, all pseudowires in VFI advertise standby mode. See the figure below.

VPLS decoupled mode is when all pseudowires in the VFI are always active and the attachment circuit state is independent of the pseudowire state. This mode provides faster switchover time when a platform does not support pseudowire status functionality, but extra flooding and multicast traffic will be dropped on the PE with standby attachment circuits. See the figure below.

Steps for Configuring Redundancy for VPLS

Perform the following task to configure redundancy for VPLS.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 vfi** *name* **manual**
4. **vpn id** *vpn-id*
5. **status decoupled**
6. **neighbor** *neighbor ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*}
7. **exit**
8. **interface port-channel** *port-channel- number*
9. **no ip address**
10. **lacp fast-switchover**
11. **lacp max-bundle** *max-bundles*
12. **exit**
13. **redundancy**
14. **interchassis group** *group-id*
15. **exit**
16. **exit**
17. **interface port-channel** *port-channel- number*
18. **service instance** *id* **ethernet** [*evc-name*]
19. **encapsulation dot1q** *vlan-id* [**,** *vlan-id*[- *vlan-id*]] [**native**]
20. **bridge-domain** *bridge-id* [**split-horizon** [**group** *group-id*]]
21. **exit**
22. **interface vlan** *vlanid*
23. **no ip address**
24. **xconnect vfi** *vfi-name*
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.

	Command or Action	Purpose
Step 3	**l2 vfi** *name* **manual** **Example:** `Router(config)# l2 vfi vfi1 manual`	Establishes a Layer 2 VFI between two separate networks and enters VFI configuration mode.
Step 4	**vpn id** *vpn-id* **Example:** `Router(config-vfi)# vpn id 100`	Sets or updates a Virtual Private Network (VPN) ID on a VPN routing and forwarding (VRF) instance.
Step 5	**status decoupled** **Example:** `Router(config-vfi)# status decoupled`	(Optional) Enables decoupled mode. The state of the attachment circuits on the user-facing Provider Edge (uPE) is decoupled from the state of the pseudowires. The mLACP controls the state of the attachment circuits.
Step 6	**neighbor** *neighbor ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} **Example:** `Router(config-vfi)# neighbor 10.1.1.1 50 encapsulation mpls`	Specifies the routers that should form a VFI connection. • Repeat this command for each neighbor.
Step 7	**exit** **Example:** `Router(config-vfi)# exit`	Exits VFI configuration mode and returns to global configuration mode.
Step 8	**interface port-channel** *port-channel- number* **Example:** `Router(config)# interface port-channel1`	Configures the port channel and enters interface configuration mode.
Step 9	**no ip address** **Example:** `Router(config-if)# no ip address`	Specifies that the VLAN interface does not have an IP address assigned to it.

Command or Action	Purpose
Step 10 **lacp fast-switchover** **Example:** `Router(config-if)# lacp fast-switchover`	Enables LACP 1-to-1 link redundancy.
Step 11 **lacp max-bundle** *max-bundles* **Example:** `Router(config-if)# lacp max-bundle 2`	Configures the max-bundle links that are connected to the PoA. The value of the *max-bundles*argument should not be less than the total number of links in the LAG that are connected to the PoA. • Determines whether the redundancy group is under DHD control, PoA control, or both. • Range is 1 to 8. Default value is 8.
Step 12 **exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 13 **redundancy** **Example:** `Router(config)# redundancy`	• Enters redundancy configuration mode.
Step 14 **interchassis group** *group-id* **Example:** `Router(config-red)# interchassis group 230`	Specifies that the port channel is an mLACP port-channel. • The *group-id* should match the configured redundancy group.
Step 15 **exit** **Example:** `Router(config-r-ic)# exit`	Exits interchassis redundancy mode.
Step 16 **exit** **Example:** `Router(config-red)# exit`	Exits redundancy configuration mode.

	Command or Action	Purpose
Step 17	**interface port-channel** *port-channel- number* **Example:** `Router(config)# interface port-channel1`	Configures the port channel and enters interface configuration mode.
Step 18	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Router(config-if)# service instance 1 ethernet`	Configures an Ethernet service instance and enters Ethernet service configuration mode.
Step 19	**encapsulation dot1q** *vlan-id* [**,** *vlan-id*[- *vlan-id*]] [**native**] **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
Step 20	**bridge-domain** *bridge-id* [**split-horizon** [**group** *group-id*]] **Example:** `Router(config-if-srv)# bridge-domain 200`	Configures the bridge domain. Binds the service instance to a bridge domain instance where *domain-number* is the identifier for the bridge domain instance.
Step 21	**exit** **Example:** `Router(config-if-srv)# exit`	Exits service instance configuration mode.
Step 22	**interface vlan** *vlanid* **Example:** `Router(config-if)# interface vlan 200`	Creates a dynamic switch virtual interface (SVI).
Step 23	**no ip address** **Example:** `Router(config-if)# no ip address`	Specifies that the VLAN interface does not have an IP address assigned to it.

	Command or Action	Purpose
Step 24	**xconnect vfi** *vfi-name* **Example:** `Router(config-if)# xconnect vfi vfi-16`	Specifies the Layer 2 VFI that you are binding to the VLAN port.
Step 25	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Configuring Hierarchical VPLS

Perform this task to configure Hierarchical VPLS (H-VPLS).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **pseudowire-class** *pw-class-name*
4. **encapsulation mpls**
5. **status peer topology dual-homed**
6. **status decoupled**
7. **exit**
8. **interface port-channel** *port-channel- number*
9. **no ip address**
10. **lacp fast-switchover**
11. **lacp max-bundle** *max-bundles*
12. **exit**
13. **redundancy**
14. **interchassis group** *group-id*
15. **exit**
16. **exit**
17. **interface port-channel** *port-channel- number*
18. **service instance** *id* **ethernet** [*evc-name*]
19. **encapsulation dot1q** *vlan-id* [**,** *vlan-id*[- *vlan-id*]] [**native**]
20. **exit**
21. **xconnect** *peer-ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} [**pw-class** *pw-class-name*] [**sequencing** {**transmit** | **receive** | **both**}]
22. **backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*] [**priority** *value*]
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.
Step 3	**pseudowire-class** *pw-class-name* **Example:** `Router(config)# pseudowire-class ether-pw`	Specifies the name of a Layer 2 pseudowire class and enters pseudowire class configuration mode.
Step 4	**encapsulation mpls** **Example:** `Router(config-pw-class)# encapsulation mpls`	Specifies that MPLS is used as the data encapsulation method for tunneling Layer 2 traffic over the pseudowire.
Step 5	**status peer topology dual-homed** **Example:** `Router(config-pw-class)# status peer topology dual-homed`	Enables the reflection of the attachment circuit status onto both the primary and secondary pseudowires. This configuration is necessary if the peer PEs are connected to a dual-homed device.
Step 6	**status decoupled** **Example:** `Router(config-pw-class)# status decoupled`	(Optional) Enables decoupled mode. The state of the attachment circuits on the uPE is decoupled from the state of the pseudowires. The mLACP controls the state of the attachment circuits.
Step 7	**exit** **Example:** `Router(config-pw-class)# exit`	Exits pseudowire class configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 8	**interface port-channel** *port-channel- number* **Example:** `Router(config)# interface port-channel1`	Configures the port channel and enters interface configuration mode.
Step 9	**no ip address** **Example:** `Router(config-if)# no ip address`	Specifies that the VLAN interface does not have an IP address assigned to it.
Step 10	**lacp fast-switchover** **Example:** `Router(config-if)# lacp fast-switchover`	Enables LACP 1-to-1 link redundancy.
Step 11	**lacp max-bundle** *max-bundles* **Example:** `Router(config-if)# lacp max-bundle 4`	Configures the max-bundle links that are connected to the PoA. The value of the *max-bundles*argument should not be less than the total number of links in the LAG that are connected to the PoA. • Determines whether the redundancy group is under DHD control, PoA control, or both. • Range is 1 to 8. Default value is 8.
Step 12	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 13	**redundancy** **Example:** `Router(config)# redundancy`	Enters redundancy configuration mode.
Step 14	**interchassis group** *group-id* **Example:** `Router(config-red)# interchassis group 230`	Specifies that the port channel is an mLACP port channel. • The *group-id* should match the configured redundancy group.

	Command or Action	Purpose
Step 15	**exit** **Example:** `Router(config-r-ic)# exit`	Exits interchassis redundancy mode.
Step 16	**exit** **Example:** `Router(config-red)# exit`	Exits redundancy configuration mode.
Step 17	**interface port-channel** *port-channel- number* **Example:** `Router(config)# interface port-channel1`	Configures the port channel and enters interface configuration mode.
Step 18	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Router(config-if)# service instance 1 ethernet`	Configures an Ethernet service instance and enters Ethernet service configuration mode.
Step 19	**encapsulation dot1q** *vlan-id* [**,** *vlan-id*[**-** *vlan-id*]] [**native**] **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
Step 20	**exit** **Example:** `Router(config-if-srv)# exit`	Exits service instance configuration mode.
Step 21	**xconnect** *peer-ip-address vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} [**pw-class** *pw-class-name*] [**sequencing** {**transmit** | **receive** | **both**}] **Example:** `Router(config-if)# xconnect 10.0.3.201 123 pw-class vlan-xconnect`	Binds an attachment circuit to a pseudowire, and configures an Any Transport over MPLS (AToM) static pseudowire.

	Command or Action	Purpose
Step 22	**backup peer** *peer-router-ip-addr vcid* [**pw-class** *pw-class-name*] [**priority** *value*] **Example:** `Router(config-if)# backup peer 10.1.1.1 123 pw-class ether-pw`	Specifies a redundant peer for a pseudowire virtual circuit.
Step 23	**end** **Example:** `Router(config-if)# end`	Returns the CLI to privileged EXEC mode.

Troubleshooting mLACP

Debugging mLACP

Use these **debug** commands for general mLACP troubleshooting.

SUMMARY STEPS

1. **enable**
2. **debug redundancy interchassis** {**all** | **application** | **error** | **event** | **monitor**}
3. **debug mpls ldp iccp**
4. **debug lacp** [**all** | **event**| **fsm**| **misc**| **multi-chassis** [**all** | **database** | **lacp-mgr** | **redundancy-group** | **user-interface**] | **packet**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug redundancy interchassis** {**all** | **application** | **error** | **event** | **monitor**} **Example:** `Router# debug redundancy interchassis all`	• Enables debugging of the interchassis redundancy manager.
Step 3	**debug mpls ldp iccp** **Example:** `Router# debug mpls ldp iccp`	• Enables debugging of the InterChassis Control Protocol (ICCP).
Step 4	**debug lacp** [**all** | **event**| **fsm**| **misc**| **multi-chassis** [**all** | **database** | **lacp-mgr** | **redundancy-group** | **user-interface**] | **packet**] **Example:** `Router# debug lacp multi-chassis all`	Enables debugging of LACP activity. • This command is run on the switch processor.

Debugging mLACP on an Attachment Circuit or EVC

Use these **debug** commands for troubleshooting mLACP on an attachment circuit or on an EVC.

SUMMARY STEPS

1. **enable**
2. **debug acircuit** {**checkpoint** | **error** | **event**}
3. **debug ethernet service** {**all** | **api** | **error** | **evc** [*evc-id*] | **ha** | **instance** [**id** *id* | **interface** *type number* | **qos**] | **interface** *type number* | **microblock** | **oam-mgr**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug acircuit** {**checkpoint** | **error** | **event**} **Example:** `Router# debug acircuit event`	Displays checkpoints, errors, and events that occur on the attachment circuits between the PE and CE routers.
Step 3	**debug ethernet service** {**all** | **api** | **error** | **evc** [*evc-id*] | **ha** | **instance** [**id** *id* | **interface** *type number* | **qos**] | **interface** *type number* | **microblock** | **oam-mgr**} **Example:** `Router# debug ethernet service all`	Enables debugging of Ethernet customer service instances.

Debugging mLACP on AToM Pseudowires

Use the **debug mpls l2transport vc** command for troubleshooting mLACP on AToM pseudowires.

SUMMARY STEPS

1. **enable**
2. **debug mpls l2transport vc** {**event** | **fsm** | **ldp** | **sss** | **status** {**event** | **fsm**}}

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	**debug mpls l2transport vc** {**event** | **fsm** | **ldp** | **sss** | **status** {**event** | **fsm**}} **Example:** `Router# debug mpls l2transport status event`	Displays information about the status of AToM virtual circuits (VCs).

Debugging Cross-Connect Redundancy Manager and Session Setup

Use the following **debug**commands to troubleshoot cross-connect, redundancy manager, and session setup.

SUMMARY STEPS

1. **enable**
2. **debug sss error**
3. **debug sss events**
4. **debug xconnect** {**error** | **event**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug sss error** **Example:** `Router# debug sss error`	Displays diagnostic information about errors that may occur during a subscriber service switch (SSS) call setup.
Step 3	**debug sss events** **Example:** `Router# debug sss event`	Displays diagnostic information about SSS call setup events.
Step 4	**debug xconnect** {**error** | **event**} **Example:** `Router# debug xconnect event`	Displays errors or events related to a cross-connect configuration.

Debugging VFI

Use the **debug vfi**command for troubleshooting a VFI.

SUMMARY STEPS

1. **enable**
2. **debug vfi** {**checkpoint** | **error** | **event** | **fsm** {**error** | **event**}}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug vfi** {**checkpoint**	**error**	**event**	**fsm** {**error**	**event**}} **Example:** `Router# debug vfi checkpoint`	Displays checkpoint information about a VFI.

Debugging the Segment Switching Manager (Switching Setup)

Use the **debug ssm**command for troubleshooting a segment switching manager (SSM).

SUMMARY STEPS

1. **enable**
2. **debug ssm** {**cm errors** | **cm events** | **fhm errors** | **fhm events** | **sm errors** | **sm events** | **sm counters** | **xdr**}

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	**debug ssm** {**cm errors** | **cm events** | **fhm errors** | **fhm events** | **sm errors** | **sm events** | **sm counters** | **xdr**} **Example:** `Router# debug ssm cm events`	Displays diagnostic information about the SSM for switched Layer 2 segments.

Debugging High Availability Features in mLACP

Use the following **debug**commands for troubleshooting High Availability features in mLACP.

SUMMARY STEPS

1. **enable**
2. **debug mpls l2transport checkpoint**
3. **debug acircuit checkpoint**
4. **debug vfi checkpoint**

DETAILED STEPS

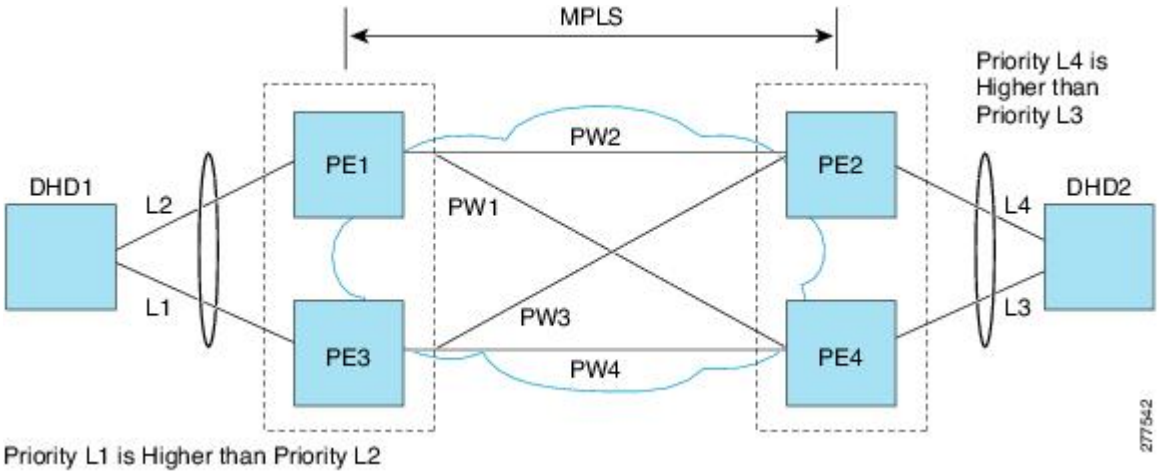
	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug mpls l2transport checkpoint** **Example:** `Router# debug mpls l2transport checkpoint`	Enables the display of AToM events when AToM is configured for nonstop forwarding/stateful switchover (NSF/SSO) and Graceful Restart.
Step 3	**debug acircuit checkpoint** **Example:** `Router# debug acircuit checkpoint`	Enables the display of attachment circuit events when AToM is configured for NSF/SSO and Graceful Restart.
Step 4	**debug vfi checkpoint** **Example:** `Router# debug vfi checkpoint`	Enables the display of VFI events when AToM is configured for NSF/SSO and Graceful Restart.

Configuration Examples for mLACP

Example Configuring VPWS

Two sample configurations for VPWS follow: one example for an active PoA and the other for a standby PoA.

The figure below shows a sample topology for a VPWS configuration.



Active PoA for VPWS

The following VPWS sample configuration is for an active PoA:

```
mpls ldp graceful-restart
mpls label protocol ldp
!
redundancy
 mode sso
 interchassis group 1
  member ip 201.0.0.1
  backbone interface Ethernet0/2
  backbone interface Ethernet1/2
  backbone interface Ethernet1/3
  monitor peer bfd
```

```
  mlacp node-id 0
!
pseudowire-class mpls-dhd
 encapsulation mpls
 status peer topology dual-homed
!
interface Loopback0
 ip address 200.0.0.1 255.255.255.255
!
interface Port-channel1
 no ip address
 lacp fast-switchover
 lacp max-bundle 1
 mlacp interchassis group 1
 hold-queue 300 in
 service instance 1 ethernet
  encapsulation dot1q 100
  xconnect 210.0.0.1 10 pw-class mpls-dhd
   backup peer 211.0.0.1 10 pw-class mpls-dhd
!
interface Ethernet0/0
 no ip address
 channel-group 1 mode active
!
interface Ethernet1/3
 ip address 10.0.0.200 255.255.255.0
 mpls ip
 bfd interval 50 min_rx 150 multiplier 3
```

Standby PoA for VPWS

The following VPWS sample configuration is for a standby PoA:

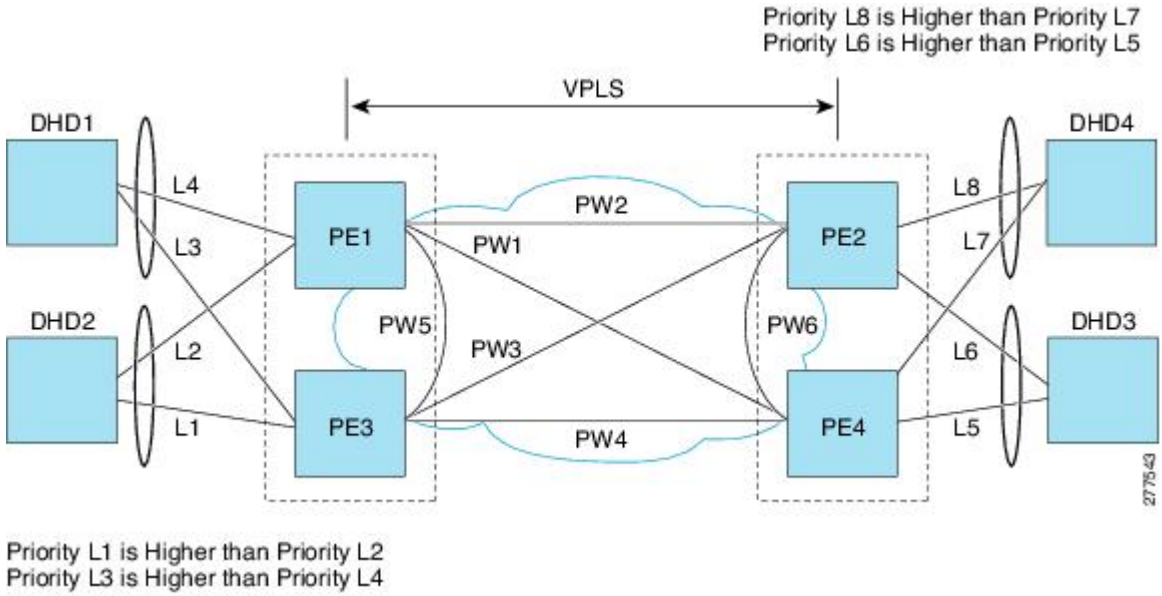
```
mpls ldp graceful-restart
mpls label protocol ldp
mpls ldp graceful-restart
mpls label protocol ldp
!
Redundancy
 mode sso
 interchassis group 1
  member ip 200.0.0.1
  backbone interface Ethernet0/2
  backbone interface Ethernet1/2
  backbone interface Ethernet1/3
  monitor peer bfd
  mlacp node-id 1
!
pseudowire-class mpls-dhd
 encapsulation mpls
 status peer topology dual-homed
!
interface Loopback0
 ip address 201.0.0.1 255.255.255.255
!
interface Port-channel1
 no ip address
 lacp fast-switchover
 lacp max-bundle 1
 mlacp lag-priority 40000
 mlacp interchassis group 1
 hold-queue 300 in
 service instance 1 ethernet
  encapsulation dot1q 100
  xconnect 210.0.0.1 10 pw-class mpls-dhd
   backup peer 211.0.0.1 10 pw-class mpls-dhd
!
interface Ethernet1/0
 no ip address
 channel-group 1 mode active
!
```

```
interface Ethernet1/3
 ip address 10.0.0.201 255.255.255.0
 mpls ip
 bfd interval 50 min_rx 150 multiplier 3
```

Example Configuring VPLS

Two sample configurations for VPLS follow: one example for an active PoA and the other for a standby PoA.

The figure below shows a sample topology for a VPLS configuration.



- Active PoA for VPLS, page 504
- Standby PoA for VPLS, page 505

Active PoA for VPLS

The following VPLS sample configuration is for an active PoA:

```
mpls ldp graceful-restart
mpls label protocol ldp
!
redundancy
 mode sso
 interchassis group 1
  member ip 201.0.0.1
  backbone interface Ethernet0/2
  monitor peer bfd
  mlacp node-id 0
!
l2 vfi VPLS_200 manual
 vpn id 10
 neighbor 210.0.0.1 encapsulation mpls
 neighbor 211.0.0.1 encapsulation mpls
 neighbor 201.0.0.1 encapsulation mpls
!
interface Loopback0
 ip address 200.0.0.1 255.255.255.255
```

```
!
interface Port-channel1
 no ip address
 lacp fast-switchover
 lacp max-bundle 1
 mlacp interchassis group 1
 service instance 1 ethernet
  encapsulation dot1q 100
  bridge-domain 200
!
interface Ethernet0/0
 no ip address
 channel-group 1 mode active
!
interface Ethernet1/3
 ip address 10.0.0.200 255.255.255.0
 mpls ip
 bfd interval 50 min_rx 150 multiplier 3
!
interface Vlan200
 no ip address
 xconnect vfi VPLS_200
```

Standby PoA for VPLS

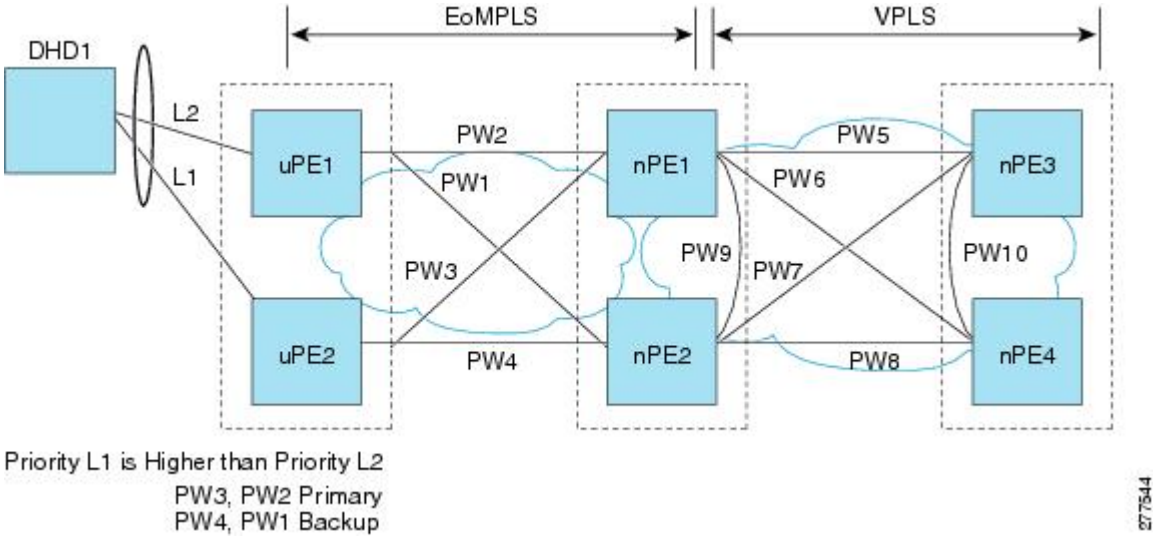
The following VPLS sample configuration is for a standby PoA:

```
mpls ldp graceful-restart
mpls label protocol ldp
!
redundancy
 interchassis group 1
  member ip 200.0.0.1
  backbone interface Ethernet0/2
  monitor peer bfd
  mlacp node-id 1
!
l2 vfi VPLS1 manual
 vpn id 10
 neighbor 210.0.0.1 encapsulation mpls
 neighbor 211.0.0.1 encapsulation mpls
 neighbor 200.0.0.1 encapsulation mpls
!
interface Loopback0
 ip address 201.0.0.1 255.255.255.255
!
interface Port-channel1
 no ip address
 lacp fast-switchover
 lacp max-bundle 1
 mlacp lag-priority 40000
 mlacp interchassis group 1
 service instance 1 ethernet
  encapsulation dot1q 100
  bridge-domain 200
!
interface Ethernet1/0
 no ip address
 channel-group 1 mode active
!
interface Ethernet1/3
 ip address 10.0.0.201 255.255.255.0
 mpls ip
 bfd interval 50 min_rx 150 multiplier 3
!
interface Vlan200
 no ip address
 xconnect vfi VPLS_200
```

Example Configuring H-VPLS

Two sample configurations for H-VPLS follow: one example for an active PoA and the other for a standby PoA.

The figure below shows a sample topology for a H-VPLS configuration.



- Active PoA for H-VPLS, page 506
- Standby PoA for H-VPLS, page 507

Active PoA for H-VPLS

The following H-VPLS sample configuration is for an active PoA:

```
mpls ldp graceful-restart
mpls label protocol ldp
!
redundancy
 mode sso
 interchassis group 1
  member ip 201.0.0.1
  backbone interface Ethernet0/2
  backbone interface Ethernet1/2
  backbone interface Ethernet1/3
  monitor peer bfd
  mlacp node-id 0
!
pseudowire-class mpls-dhd
 encapsulation mpls
 status peer topology dual-homed
!
interface Loopback0
 ip address 200.0.0.1 255.255.255.255
!
interface Port-channel1
 no ip address
 lacp fast-switchover
 lacp max-bundle 1
 mlacp interchassis group 1
 hold-queue 300 in
 service instance 1 ethernet
```

```
  encapsulation dot1q 100
  xconnect 210.0.0.1 10 pw-class mpls-dhd
   backup peer 211.0.0.1 10 pw-class mpls-dhd
!
interface Ethernet0/0
 no ip address
 channel-group 1 mode active
!
interface Ethernet1/3
 ip address 10.0.0.200 255.255.255.0
 mpls ip
 bfd interval 50 min_rx 150 multiplier 3
```

Standby PoA for H-VPLS

The following H-VPLS sample configuration is for a standby PoA:

```
mpls ldp graceful-restart
mpls label protocol ldp
!
Redundancy
 mode sso
 interchassis group 1
  member ip 200.0.0.1
  backbone interface Ethernet0/2
  backbone interface Ethernet1/2
  backbone interface Ethernet1/3
  monitor peer bfd
  mlacp node-id 1
!
pseudowire-class mpls-dhd
 encapsulation mpls
 status peer topology dual-homed
!
interface Loopback0
 ip address 201.0.0.1 255.255.255.255
!
interface Port-channel1
 no ip address
 lacp fast-switchover
 lacp max-bundle 1
 mlacp lag-priority 40000
 mlacp interchassis group 1
 hold-queue 300 in
 service instance 1 ethernet
  encapsulation dot1q 100
  xconnect 210.0.0.1 10 pw-class mpls-dhd
   backup peer 211.0.0.1 10 pw-class mpls-dhd
!
interface Ethernet1/0
 no ip address
 channel-group 1 mode active
!
interface Ethernet1/3
 ip address 10.0.0.201 255.255.255.0
 mpls ip
 bfd interval 50 min_rx 150 multiplier 3
```

Example Verifying VPWS on an Active PoA

The following **show** commands can be used to display statistics and configuration parameters to verify the operation of the mLACP feature on an active PoA:

- show lacp multichassis group, page 508
- show lacp multichassis port-channel, page 508
- show mpls ldp iccp, page 509
- show mpls l2transport, page 509

- show etherchannel summary, page 509
- show etherchannel number port-channel, page 509
- show lacp internal, page 510

show lacp multichassis group

Use the **show lacp multichassis group** command to display the interchassis redundancy group value and the operational LACP parameters.

```
Router# show lacp multichassis group 100
Interchassis Redundancy Group 100
Operational LACP Parameters:
RG State:     Synchronized
System-Id:    200.000a.f331.2680
ICCP Version: 0
Backbone Uplink Status: Connected
Local Configuration:
Node-id:   0
System-Id: 200.000a.f331.2680
Peer Information:
State:        Up
Node-id:      7
System-Id:    2000.0014.6a8b.c680
ICCP Version: 0
State Flags: Active            - A
             Standby           - S
             Down              - D
             AdminDown         - AD
             Standby Reverting - SR
             Unknown           - U

mLACP Channel-groups
Channel    State      Priority     Active Links   Inactive Links
 Group   Local/Peer  Local/Peer     Local/Peer      Local/Peer
   1        A/S     28000/32768        4/4             0/0
```

show lacp multichassis port-channel

Use the **show lacp multichassis port-channel** command to display the interface port-channel value channel group, LAG state, priority, inactive links peer configuration, and standby links.

```
Router# show lacp multichassis port-channel1
Interface Port-channel1
Local Configuration:
Address: 000a.f331.2680
Channel Group: 1
State: Active
LAG State: Up
Priority: 28000
Inactive Links: 0
Total Active Links: 4
           Bundled: 4
          Selected: 4
           Standby: 0
        Unselected: 0
Peer Configuration:
Interface: Port-channel1
Address: 0014.6a8b.c680
Channel Group: 1
State: Standby
LAG State: Up
Priority: 32768
Inactive Links: 0
Total Active Links: 4
                          Bundled: 0
```

```
          Selected: 0
           Standby: 4
        Unselected: 0
```

show mpls ldp iccp

Use the **show mpls ldp iccp** command to display the LDP session and ICCP state information.

```
Router# show mpls ldp iccp

ICPM RGID Table
  iccp:
    rg_id: 100, peer addr: 172.3.3.3
    ldp_session 0x3, client_id 0
    iccp state: ICPM_ICCP_CONNECTED
    app type: MLACP
        app state: ICPM_APP_CONNECTED, ptcl ver: 0
ICPM RGID Table total ICCP sessions: 1
ICPM LDP Session Table
  iccp:
    rg_id: 100, peer addr: 172.3.3.3
    ldp_session 0x3, client_id 0
    iccp state: ICPM_ICCP_CONNECTED
    app type: MLACP
        app state: ICPM_APP_CONNECTED, ptcl ver: 0
ICPM LDP Session Table total ICCP sessions: 1
```

show mpls l2transport

Use the **show mpls l2transport** command to display the local interface and session details, destination address, and status.

```
Router# show mpls l2transport vc 2
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
Po1            Eth VLAN 2                 172.2.2.2       2          UP
Po1            Eth VLAN 2                 172.4.4.4       2          STANDBY
```

show etherchannel summary

Use the **show etherchannel summary** command to display the status and identity of the mLACP member links.

```
Router# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port
Number of channel-groups in use: 2
Number of aggregators:           2
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(RU)         LACP      Gi2/9(P)    Gi2/20(P)   Gi2/31(P)
```

show etherchannel number port-channel

Use the **show etherchannel number port-channel** command to display the status and identity of the EtherChannel and and port channel.

```
Router# show etherchannel 51 port-c

                Port-channels in the group:
                ----------------------

Port-channel: Po51    (Primary Aggregator)

------------

Age of the Port-channel   = 0d:02h:25m:23s
Logical slot/port   = 14/11          Number of ports = 2
HotStandBy port = null
Passive port list   = Gi9/15 Gi9/16
Port state          = Port-channel L3-Ag Ag-Inuse
Protocol            =   LACP
Fast-switchover     = enabled
Direct Load Swap    = disabled

Ports in the Port-channel:

Index   Load    Port      EC state        No of bits
------+------+--------+------------------+-----------
  0     55     Gi9/15     mLACP-stdby   4
  1     AA     Gi9/16     mLACP-stdby   4

Time since last port bundled:    0d:01h:03m:39s    Gi9/16
Time since last port Un-bundled: 0d:01h:03m:40s    Gi9/16

Last applied Hash Distribution Algorithm: Fixed Channel-group Iedge Counts:
-------------------------:
Access ref count       : 0
Iedge session count    : 0
```

show lacp internal

Use the **show lacp internal**command to display the device, port, and member- link information.

```
Router# show lacp internal
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode
Channel group 1
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Gi2/9     SA      bndl-act  28000         0x1       0x1     0x820A      0x3D
Gi2/20    SA      bndl-act  28000         0x1       0x1     0x8215      0x3D
Gi2/31    SA      bndl-act  28000         0x1       0x1     0x8220      0x3D
Gi2/40    SA      bndl-act  28000         0x1       0x1     0x8229      0x3D
Peer (MLACP-PE3) mLACP member links
Gi3/11    FA      hot-sby   32768         0x1       0x1     0xF30C      0x5
Gi3/21    FA      hot-sby   32768         0x1       0x1     0xF316      0x5
Gi3/32    FA      hot-sby   32768         0x1       0x1     0xF321      0x7
Gi3/2     FA      hot-sby   32768         0x1       0x1     0xF303      0x7
```

Example Verifying VPWS on a Standby PoA

The following **show** commands can be used to display statistics and configuration parameters to verify the operation of the mLACP feature on a standby PoA:

- show lacp multichassis group, page 511
- show lacp multichassis portchannel, page 511
- show mpls ldp iccp, page 512
- show mpls l2transport, page 512

- show etherchannel summary, page 512
- show lacp internal, page 512

show lacp multichassis group

Use the **show lacp multichassis group** command to display the LACP parameters, local configuration, status of the backbone uplink, peer information, node ID, channel, state, priority active, and inactive links.

```
Router# show lacp multichassis group 100
Interchassis Redundancy Group 100
Operational LACP Parameters:
RG State:     Synchronized
System-Id:    200.000a.f331.2680
ICCP Version: 0
Backbone Uplink Status: Connected
Local Configuration:
Node-id:   7
System-Id: 2000.0014.6a8b.c680
Peer Information:
State:        Up
Node-id:      0
System-Id:    200.000a.f331.2680
ICCP Version: 0
State Flags: Active            - A
             Standby           - S
             Down              - D
             AdminDown         - AD
             Standby Reverting - SR
             Unknown           - U

mLACP Channel-groups
Channel    State      Priority     Active Links   Inactive Links
 Group   Local/Peer  Local/Peer     Local/Peer      Local/Peer
   1        S/A     32768/28000        4/4             0/0
```

show lacp multichassis portchannel

Use the **show lacp multichassis portchannel** command to display the interface port-channel value channel group, LAG state, priority, inactive links peer configuration, and standby links.

```
Router# show lacp multichassis port-channel1
Interface Port-channel1
Local Configuration:
Address: 0014.6a8b.c680
Channel Group: 1
State: Standby
LAG State: Up
Priority: 32768
Inactive Links: 0
Total Active Links: 4
           Bundled: 0
          Selected: 0
           Standby: 4
        Unselected: 0
Peer Configuration:
Interface: Port-channel1
Address: 000a.f331.2680
Channel Group: 1
State: Active
LAG State: Up
Priority: 28000
Inactive Links: 0
Total Active Links: 4
                          Bundled: 4
          Selected: 4
```

```
          Standby: 0
        Unselected: 0
```

show mpls ldp iccp

Use the **show mpls ldp iccp**command to display the LDP session and ICCP state information.

```
Router# show mpls ldp iccp
ICPM RGID Table
  iccp:
    rg_id: 100, peer addr: 172.1.1.1
    ldp_session 0x2, client_id 0
    iccp state: ICPM_ICCP_CONNECTED
    app type: MLACP
        app state: ICPM_APP_CONNECTED, ptcl ver: 0
ICPM RGID Table total ICCP sessions: 1
ICPM LDP Session Table
  iccp:
    rg_id: 100, peer addr: 172.1.1.1
    ldp_session 0x2, client_id 0
    iccp state: ICPM_ICCP_CONNECTED
    app type: MLACP
        app state: ICPM_APP_CONNECTED, ptcl ver: 0
ICPM LDP Session Table total ICCP sessions: 1
```

show mpls l2transport

Use the **show mpls l2transport** command to display the local interface and session details, destination address, and status.

```
Router# show mpls l2transport vc 2
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
Po1            Eth VLAN 2                 172.2.2.2       2          STANDBY
Po1            Eth VLAN 2                 172.4.4.4       2          STANDBY
```

show etherchannel summary

Use the **show etherchannel summary** command to display the status and identity of the mLACP member links.

```
Router# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port
Number of channel-groups in use: 2
Number of aggregators:           2
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(RU)         LACP      Gi3/2(P)    Gi3/11(P)   Gi3/21(P)
                                 Gi3/32(P)
```

show lacp internal

Use the **show lacp internal** command to display the device, port, and member-link information.

```
Router# show lacp 1 internal
```

```
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode
Channel group 1
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Gi3/2     FA      bndl-sby  32768         0x1       0x1     0xF303      0x7
Gi3/11    FA      bndl-sby  32768         0x1       0x1     0xF30C      0x5
Gi3/21    FA      bndl-sby  32768         0x1       0x1     0xF316      0x5
Gi3/32    FA      bndl-sby  32768         0x1       0x1     0xF321      0x7
Peer (MLACP-PE1) mLACP member links
Gi2/20    SA      bndl      28000         0x1       0x1     0x8215      0x3D
Gi2/31    SA      bndl      28000         0x1       0x1     0x8220      0x3D
Gi2/40    SA      bndl      28000         0x1       0x1     0x8229      0x3D
Gi2/9     SA      bndl      28000         0x1       0x1     0x820A      0x3D
```

Example Verifying VPLS on an Active PoA

The following **show** commands can be used to display statistics and configuration parameters to verify the operation of the mLACP feature on an active PoA:

- show lacp multichassis group, page 513
- show lacp multichassis port-channel, page 513
- show mpls ldp iccp, page 514
- show mpls l2transport, page 514
- show etherchannel summary, page 515
- show lacp internal, page 515

show lacp multichassis group

Use the **show lacp multichassis group** command to display the LACP parameters, local configuration, status of the backbone uplink, peer information, node ID, channel, state, priority active, and inactive links.

```
Router# show lacp multichassis group 100
Interchassis Redundancy Group 100
Operational LACP Parameters:
RG State:     Synchronized
System-Id:    200.000a.f331.2680
ICCP Version: 0
Backbone Uplink Status: Connected
Local Configuration:
Node-id:   0
System-Id: 200.000a.f331.2680
Peer Information:
State:        Up
Node-id:      7
System-Id:    2000.0014.6a8b.c680
ICCP Version: 0
State Flags: Active            - A
             Standby           - S
             Down              - D
             AdminDown         - AD
             Standby Reverting - SR
             Unknown           - U

mLACP Channel-groups
Channel    State      Priority     Active Links   Inactive Links
 Group   Local/Peer  Local/Peer     Local/Peer      Local/Peer
   1        A/S     28000/32768        4/4             0/0
```

show lacp multichassis port-channel

Use the **show lacp multichassis port-channel** command to display the interface port-channel value channel group, LAG state, priority, inactive links peer configuration, and standby links.

```
Router# show lacp multichassis port-channel1
Interface Port-channel1
Local Configuration:
Address: 000a.f331.2680
Channel Group: 1
State: Active
LAG State: Up
Priority: 28000
Inactive Links: 0
Total Active Links: 4
           Bundled: 4
          Selected: 4
           Standby: 0
        Unselected: 0
Peer Configuration:
Interface: Port-channel1
Address: 0014.6a8b.c680
Channel Group: 1
State: Standby
LAG State: Up
Priority: 32768
Inactive Links: 0
Total Active Links: 4
                          Bundled: 0
          Selected: 0
           Standby: 4
        Unselected: 0
```

show mpls ldp iccp

Use the **show mpls ldp iccp** command to display the LDP session and ICCP state information.

```
Router# show mpls ldp iccp
ICPM RGID Table
  iccp:
    rg_id: 100, peer addr: 172.3.3.3
    ldp_session 0x3, client_id 0
    iccp state: ICPM_ICCP_CONNECTED
    app type: MLACP
        app state: ICPM_APP_CONNECTED, ptcl ver: 0
ICPM RGID Table total ICCP sessions: 1
ICPM LDP Session Table
  iccp:
    rg_id: 100, peer addr: 172.3.3.3
    ldp_session 0x3, client_id 0
    iccp state: ICPM_ICCP_CONNECTED
    app type: MLACP
        app state: ICPM_APP_CONNECTED, ptcl ver: 0
ICPM LDP Session Table total ICCP sessions: 1
```

show mpls l2transport

Use the **show mpls l2transport** command to display the local interface and session details, destination address, and the status.

```
Router# show mpls l2transport vc 4000
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
VFI VPLS       VFI                        172.2.2.2       4000       UP
VFI VPLS       VFI                        172.4.4.4      4000       UP
```

show etherchannel summary

Use the **show etherchannel summary** command to display the status and identity of the mLACP member links.

```
Router# show etherchannel summary
Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port
Number of channel-groups in use: 2
Number of aggregators:           2
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(RU)         LACP      Gi2/9(P)    Gi2/20(P)   Gi2/31(P)
                                 Gi2/40(P)
```

show lacp internal

Use the **show lacp internal** command to display the device, port, and member-link information.

```
Router# show lacp internal
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode
Channel group 1
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Gi2/9     SA      bndl-act  28000         0x1       0x1     0x820A      0x3D
Gi2/20    SA      bndl-act  28000         0x1       0x1     0x8215      0x3D
Gi2/31    SA      bndl-act  28000         0x1       0x1     0x8220      0x3D
Gi2/40    SA      bndl-act  28000         0x1       0x1     0x8229      0x3D
Peer (MLACP-PE3) mLACP member links
Gi3/11    FA      hot-sby   32768         0x1       0x1     0xF30C      0x5
Gi3/21    FA      hot-sby   32768         0x1       0x1     0xF316      0x5
Gi3/32    FA      hot-sby   32768         0x1       0x1     0xF321      0x7
Gi3/2     FA      hot-sby   32768         0x1       0x1     0xF303      0x7
```

Example Verifying VPLS on a Standby PoA

The **show** commands in this section can be used to display statistics and configuration parameters to verify the operation of the mLACP feature:

- show lacp multichassis group, page 515
- show lacp multichassis portchannel, page 516
- show mpls ldp iccp, page 516
- show mpls l2transport vc 2, page 517
- showetherchannelsummary, page 517
- show lacp internal, page 517

show lacp multichassis group

Use the **show lacp multichassis group** *interchassis group number* command to display the LACP parameters, local configuration, status of the backbone uplink, peer information, node ID, channel, state, priority, active, and inactive links.

```
Router# show lacp multichassis group 100
Interchassis Redundancy Group 100
Operational LACP Parameters:
RG State:     Synchronized
System-Id:    200.000a.f331.2680
ICCP Version: 0
Backbone Uplink Status: Connected
Local Configuration:
Node-id:   7
System-Id: 2000.0014.6a8b.c680
Peer Information:
State:        Up
Node-id:      0
System-Id:    200.000a.f331.2680
ICCP Version: 0
State Flags: Active            - A
             Standby           - S
             Down              - D
             AdminDown         - AD
             Standby Reverting - SR
             Unknown           - U

mLACP Channel-groups
Channel    State      Priority     Active Links   Inactive Links
 Group   Local/Peer  Local/Peer     Local/Peer      Local/Peer
   1        S/A     32768/28000        4/4             0/0
```

show lacp multichassis portchannel

Use the **show lacp multichassis portchannel** command to display the interface port-channel value channel group, LAG state, priority, inactive links peer configuration, and standby links.

```
Router# show lacp multichassis port-channel1
Interface Port-channel1
Local Configuration:
Address: 0014.6a8b.c680
Channel Group: 1
State: Standby
LAG State: Up
Priority: 32768
Inactive Links: 0
Total Active Links: 4
           Bundled: 0
          Selected: 0
           Standby: 4
        Unselected: 0
Peer Configuration:
Interface: Port-channel1
Address: 000a.f331.2680
Channel Group: 1
State: Active
LAG State: Up
Priority: 28000
Inactive Links: 0
Total Active Links: 4
                            Bundled: 4
          Selected: 4
           Standby: 0
        Unselected: 0
```

show mpls ldp iccp

Use the **show mpls ldp iccp** command to display the LDP session and ICCP state information.

```
Router# show mpls ldp iccp
ICPM RGID Table
  iccp:
    rg_id: 100, peer addr: 172.1.1.1
    ldp_session 0x2, client_id 0
    iccp state: ICPM_ICCP_CONNECTED
    app type: MLACP
        app state: ICPM_APP_CONNECTED, ptcl ver: 0
ICPM RGID Table total ICCP sessions: 1
ICPM LDP Session Table
  iccp:
    rg_id: 100, peer addr: 172.1.1.1
    ldp_session 0x2, client_id 0
    iccp state: ICPM_ICCP_CONNECTED
    app type: MLACP
        app state: ICPM_APP_CONNECTED, ptcl ver: 0
ICPM LDP Session Table total ICCP sessions: 1
```

show mpls l2transport vc 2

Use the **show mpls l2transport** command to display the local interface and session details, destination address, and status.

```
Router# show mpls l2transport vc 2
Local intf     Local circuit              Dest address    VC ID      Status
-------------  -------------------------- --------------- ---------- ----------
VFI VPLS       VFI                        172.2.2.2       4000       UP
VFI VPLS       VFI                                                    172.4.4.4
4000           UP
```

showetherchannelsummary

Use the **show etherchannel summary** command to display the status and identity of the mLACP member links.

```
Router# show etherchannel summary

Flags:  D - down        P - bundled in port-channel
        I - stand-alone s - suspended
        H - Hot-standby (LACP only)
        R - Layer3      S - Layer2
        U - in use      f - failed to allocate aggregator
        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port
Number of channel-groups in use: 2
Number of aggregators:           2
Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(RU)         LACP      Gi3/2(P)    Gi3/11(P)   Gi3/21(P)
                                 Gi3/32(P)
```

show lacp internal

Use the **show lacp internal** command to display the device, port, and member- link information.

```
Router# show lacp 1 internal

Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode
```

```
Channel group 1
                            LACP port     Admin     Oper    Port        Port
Port      Flags   State     Priority      Key       Key     Number      State
Gi3/2     FA      bndl-sby  32768         0x1       0x1     0xF303      0x7
Gi3/11    FA      bndl-sby  32768         0x1       0x1     0xF30C      0x5
Gi3/21    FA      bndl-sby  32768         0x1       0x1     0xF316      0x5
Gi3/32    FA      bndl-sby  32768         0x1       0x1     0xF321      0x7
Peer (MLACP-PE1) mLACP member links
Gi2/20    SA      bndl      28000         0x1       0x1     0x8215      0x3D
Gi2/31    SA      bndl      28000         0x1       0x1     0x8220      0x3D
Gi2/40    SA      bndl      28000         0x1       0x1     0x8229      0x3D
Gi2/9     SA      bndl      28000         0x1       0x1     0x820A      0x3D
```

Additional References

Related Documents

Related Topic	Document Title
Carrier Ethernet configurations	*Cisco IOS Carrier Ethernet Configuration Guide , Release 12.2SR*
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

Standards

Standard	Title
IEEE 802.3ad	*Link Aggregation Control Protocol*
IEEE 802.1ak	*Multiple Registration Protocol*

MIBs

MIB	MIBs Link
• Cisco-LAG-MIB • IEEE 802.3ad-MIB • IEEE8023-LAG-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 4762	*Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*
RFC 4447	*Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*

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 mLACP

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 23 Feature Information for mLACP

Feature Name	Releases	Feature Information
Multichassis LACP (mLACP)	12.2(33)SRE 15.0(1)S	Cisco's mLACP feature addresses the need for interchassis redundancy mechanisms when a carrier wants to dual home a device to two upstream PoAs for redundancy. The mLACP feature enhances the 802.3ad LACP implementation to meet this requirement. The following commands were introduced or modified: **backbone interface**, **debug acircuit checkpoint**, **debug lacp**, **ethernet mac-flush mirp notification**, **interchassis group**, **lacp failover**, **lacp max-bundle**, **lacp min-bundle**, **member ip**, **mlacp interchassis group**, **mlacp lag-priority**, **mlacp node-id**, **mlacp system-mac**, **mlacp system-priority**, **monitor peer bfd**, **redundancy**, **show ethernet service instance interface port-channel**, **show ethernet service instance id mac-tunnel**, **show lacp**, **status decoupled**, **status peer topology dual-homed**.

Glossary

active attachment circuit—The link that is actively forwarding traffic between the DHD and the active PoA.

active PW—The pseudowire that is forwarding traffic on the active PoA.

BD—bridge domain.

BFD—bidirectional forwarding detection.

DHD—dual-homed device. A node that is connected to two switches over a multichassis link aggregation group for the purpose of redundancy.

DHN—dual-homed network. A network that is connected to two switches to provide redundancy.

H-VPLS—Hierarchical Virtual Private LAN Service.

ICC—Interchassis Communication Channel.

ICCP—Interchassis Communication Protocol.

ICPM—Interchassis Protocol Manager.

ICRM—Interchassis Redundancy Manager.

LACP—Link Aggregation Control Protocol.

LAG—link aggregation group.

LDP—Link Distribution Protocol.

MCEC—Multichassis EtherChannel.

mLACP—Multichassis LACP.

PoA—point of attachment. One of a pair of switches running multichassis link aggregation group with a DHD.

PW-RED—pseudowire redundancy.

standby attachment circuit—The link that is in standby mode between the DHD and the standby PoA.

standby PW—The pseudowire that is in standby mode on either an active or a standby PoA.

uPE—user-facing Provider Edge.

VPLS—Virtual Private LAN Service.

VPWS—Virtual Private Wire Service.