Carrier Ethernet Configuration Guide, Cisco IOS Release 15S

First Published: November 06, 2012

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

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.

CONTENTS

CHAPTER 1

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 caveats and feature information, see Bug Search Tool and 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 module.

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 Implementation of Ethernet OAM

The Cisco 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 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, when an interface is shut down. 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.

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 Switch Processor (RSP) (a standby RSP that has the same software image as the active RSP and supports synchronization of line card, protocol, and application state information between RSPs 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 CFM and 802.3ah. This status information is used to either stop traffic or switch to backup paths when an EVC is down. Metro Ethernet clients (for example, CFM and 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 RSPs are transparent to clients.

Cisco infrastructure provides various component application program interfaces (APIs) for clients that are helpful in maintaining a hot standby RSP. Metro Ethernet HA clients (such as, 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

- Elimination of network downtime for Cisco 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 Stateful Switchover (SSO) and Nonstop Forwarding (NSF) are both supported in Ethernet OAM and are automatically enabled. A switchover from an active to a standby Route Switch Processor (RSP) occurs when the active RSP 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 RSP switchover.

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

ISSU Support in 802.3ah OAM

Cisco In-Service Software Upgrades (ISSUs) allow you to perform a Cisco 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 Route Switch Processor (RSP), 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 RSP to standby RSP 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 “Performing an In Service Software Upgrade” module of the *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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	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:** `Device(config-if)# ethernet oam`	Enables Ethernet OAM.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Returns 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	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:** `Device(config-if)# ethernet oam`	Enables Ethernet OAM.

	Command or Action	Purpose
Step 5	**no ethernet oam link-monitor supported** **Example:** `Device(config-if)# no ethernet oam link-monitor supported`	Disables link monitoring on the interface.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Returns 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ethernet oam link-monitor supported** **Example:** `Device(config-if)# ethernet oam link-monitor supported`	Enables link monitoring on the interface.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Returns 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	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:** `Device(config-if)# ethernet oam`	Enables Ethernet OAM.
Step 5	**no ethernet oam link-monitor on** **Example:** `Device(config-if)# no ethernet oam link-monitor on`	Stops link monitoring operations.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Returns 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Specifies an interface and enters interface configuration mode.
Step 4	**ethernet oam link-monitor on** **Example:** `Device(config-if)# ethernet oam link-monitor on`	Starts link monitoring operations.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Returns 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	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:** `Device(config-if)# ethernet oam`	Enables Ethernet OAM.

	Command or Action	Purpose
Step 5	**ethernet oam link-monitor high-threshold action error-disable-interface** **Example:** `Device(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:
	`Device(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:
	`Device(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.

Step 8	**ethernet oam link-monitor frame-seconds** {**threshold** {**high** {**none** | *high-frames*} | **low** *low-frames*} | **window** *milliseconds*}
	Example:
	`Device(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:
	`Device(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:
	`Device(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.

	Command or Action	Purpose

Step 11	**ethernet oam link-monitor symbol-period** {**threshold** {**high** {**none** | *high-symbols*} | **low** *low-symbols*} | **window** *symbols*}
	Example:
	`Device(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:** `Device(config-if)# exit`	Returns to global configuration mode.

Example

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**template** *template-name* **Example:** `Device(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:
	`Device(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:
	`Device(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:
	`Device(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:** `Device(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:
	`Device(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:
	`Device(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.

Step 10	**ethernet oam link-monitor frame-seconds** {**threshold** {**high** {**none** | *high-frames*} | **low** *low-frames*} | **window** *milliseconds*}	Configures a period of time in which error frames are counted.

	Command or Action	Purpose
	Example: `Device(config-template)# ethernet oam link-monitor frame-seconds window 699`	
Step 11	**exit** **Example:** `Device(config-template)# exit`	Returns to global configuration mode.
Step 12	**interface** *type number* **Example:**	Identifies the interface on which to use the template and enters interface configuration mode.
Step 13	**source template** *template-name* **Example:** `Device(config-if)# source template oam-temp`	Applies to the interface the options configured in the template.
Step 14	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 15	**exit** **Example:** `Device(config)# exit`	Returns to privileged EXEC mode.
Step 16	**show running-config** **Example:** `Device# 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-disable-interface**}
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	Enters interface configuration mode.

Step 4	**ethernet oam remote-failure** {**critical-event** | **dying-gasp** | **link-fault**} **action** {**error-disable-interface**}
	Example:
	`Device(config-if)# ethernet oam remote-failure critical-event action error-disable-interface`
		Sets the interface to the blocking state when a critical event occurs.

Step 5	**exit** **Example:** `Device(config-if)# exit`	Returns 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.
!
Device(config)# template oam
Device(config-template)# ethernet oam link-monitor symbol-period threshold low 10
Device(config-template)# ethernet oam link-monitor symbol-period threshold high 100
Device(config-template)# ethernet oam link-monitor frame window 100
Device(config-template)# ethernet oam link-monitor frame threshold low 10
Device(config-template)# ethernet oam link-monitor frame threshold high 100
Device(config-template)# ethernet oam link-monitor frame-period window 100
Device(config-template)# ethernet oam link-monitor frame-period threshold low 10
Device(config-template)# ethernet oam link-monitor frame-period threshold high 100
Device(config-template)# ethernet oam link-monitor frame-seconds window 1000
Device(config-template)# ethernet oam link-monitor frame-seconds threshold low 10
Device(config-template)# ethernet oam link-monitor frame-seconds threshold high 100
Device(config-template)# ethernet oam link-monitor receive-crc window 100
Device(config-template)# ethernet oam link-monitor receive-crc threshold high 100
Device(config-template)# ethernet oam link-monitor transmit-crc window 100
Device(config-template)# ethernet oam link-monitor transmit-crc threshold high 100
Device(config-template)# ethernet oam remote-failure dying-gasp action error-disable-interface
Device(config-template)# exit
!
! Enable Ethernet OAM on the CE interface
!
Device(config)#
Device(config-if)# ethernet oam
!
! Apply the global OAM template named "oam" to the interface.
!
Device(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.
!
Device(config-if)# ethernet oam link-monitor receive-crc threshold high none
!
! Enable Ethernet OAM on the PE interface
!
Device(config)#
Device(config-if)# ethernet oam
!
! Apply the global OAM template named "oam" to the interface.
!
Device(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 , 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.

```
Device# 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:

```
Device#

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:

```
Device#

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).

```
Device#

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.

```
Device# 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” module in the *Carrier Ethernet Configuration Guide*
NSF SSO Support in 802.3ah OAM	“Configuring Stateful Switchover” module in the *High Availability Configuration Guide* and “Configuring Nonstop Forwarding” in the *High Availability Configuration Guide*
ISSU Support in 802.3ah OAM	“Configuring In Service Software Upgrades" module in the *High Availability 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
Configuring CFM over an EFP Interface with the Cross Connect feature on the Cisco ASR 903 Router	Configuring the CFM over EFP Interface with Cross Connect Feature
Configuring Ethernet Virtual Connections on the Cisco ASR 903 Router	Configuring Ethernet Virtual Connections on the Cisco ASR 903 Router

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*

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	12.4(15)T	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 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)**.

CHAPTER 2

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 caveats and feature information, see Bug Search Tool and 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 module.

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.

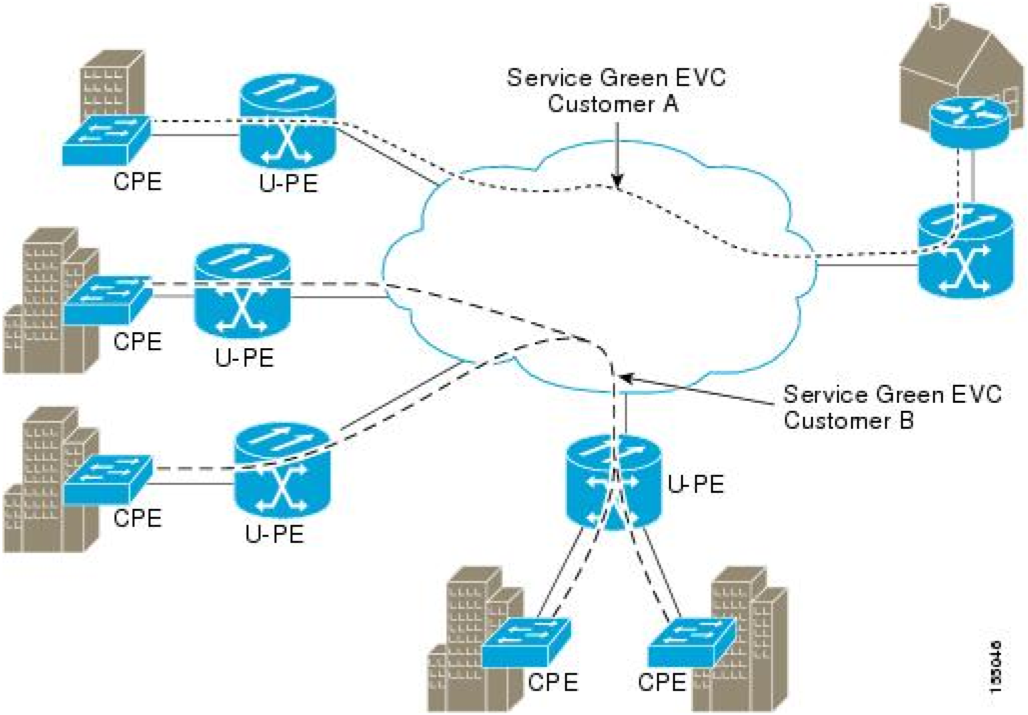
Benefits of IEEE CFM

- 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.



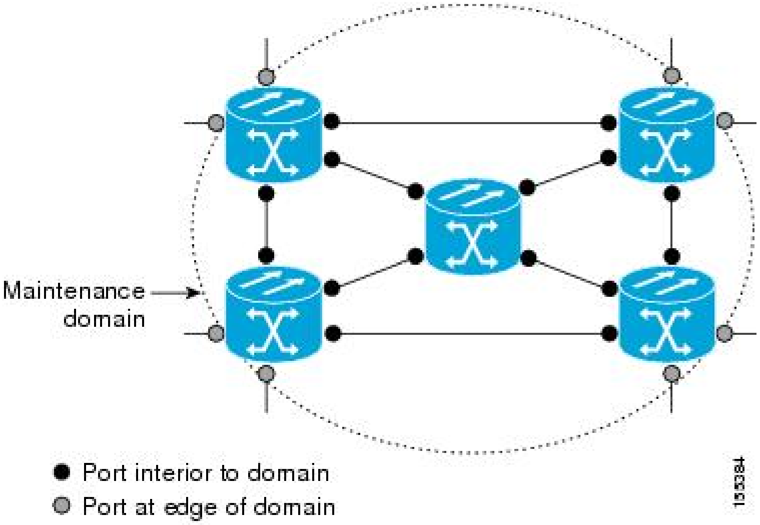
Maintenance Association

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.

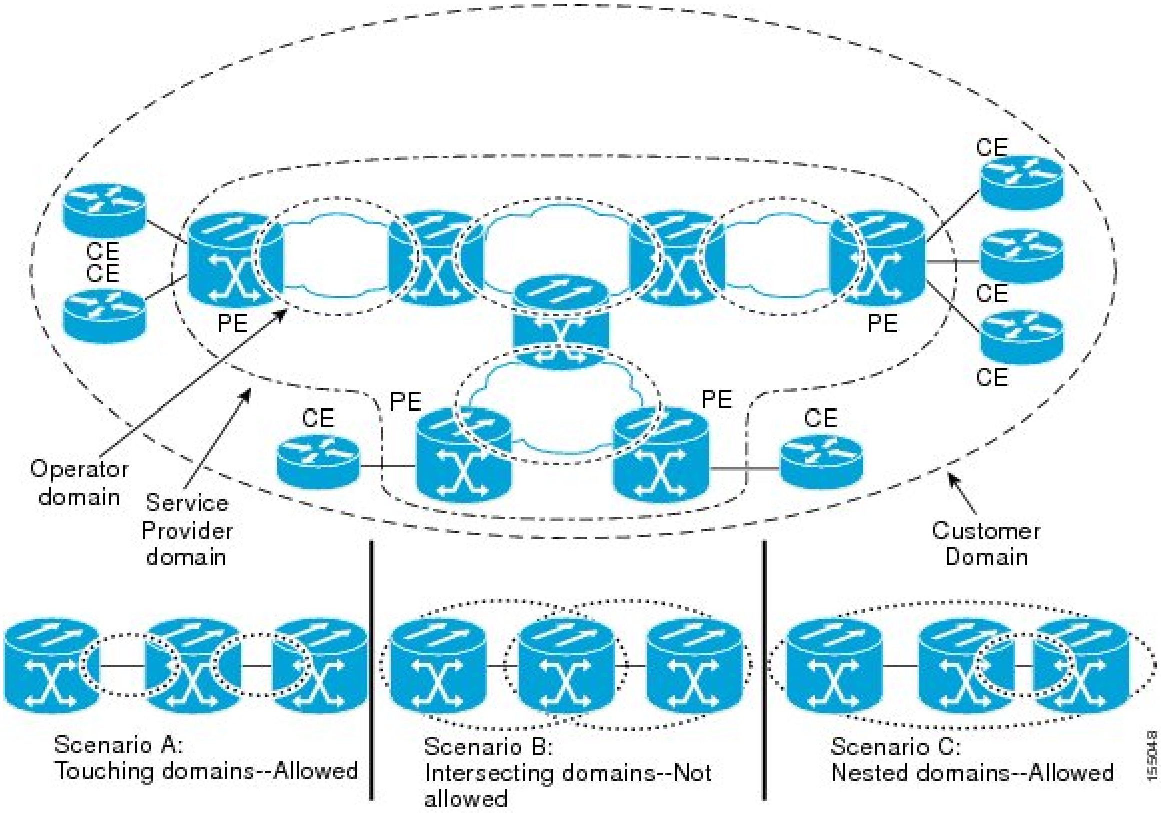
Network designers decide on domains and configurations.

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.


CE
CE
CE
CE
CE
CE
PE
PE
PE
PE
CE
CE
Operator domain
Service Provider domain
Customer Domain
Scenario A: Touching domains--Allowed
Scenario B: Intersecting domains--Not allowed
Scenario C: Nested domains--Allowed
155048

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

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.

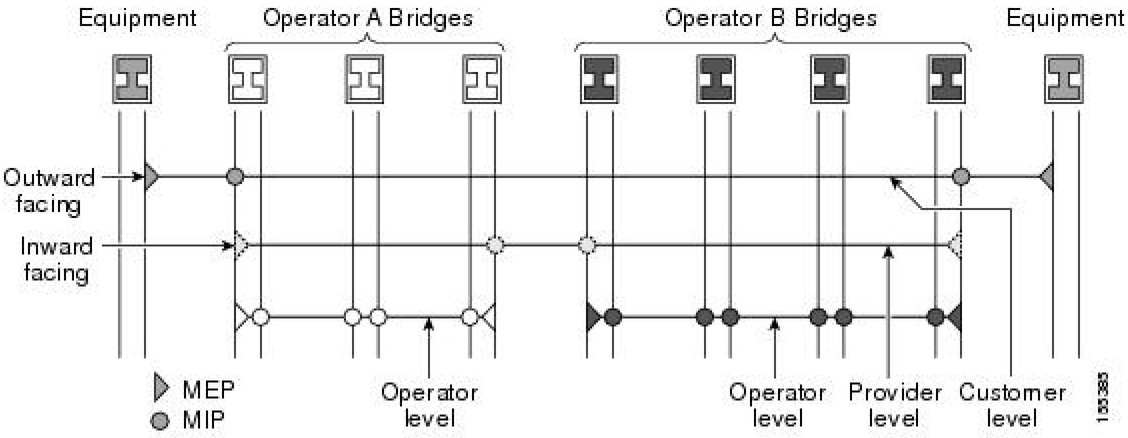
MIPs are defined as two MIP half functions (MHFs): An Up MHF that resides above the port filtering entities and a Down MHF that resides below the port filtering entities. The same configuration parameters and characteristics apply to both MHFs of a MIP, as follows:

- 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 postprovisioning service verification between dynamically discovered MEPs (via continuity check messages 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

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

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

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.

Before You Begin

- 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.

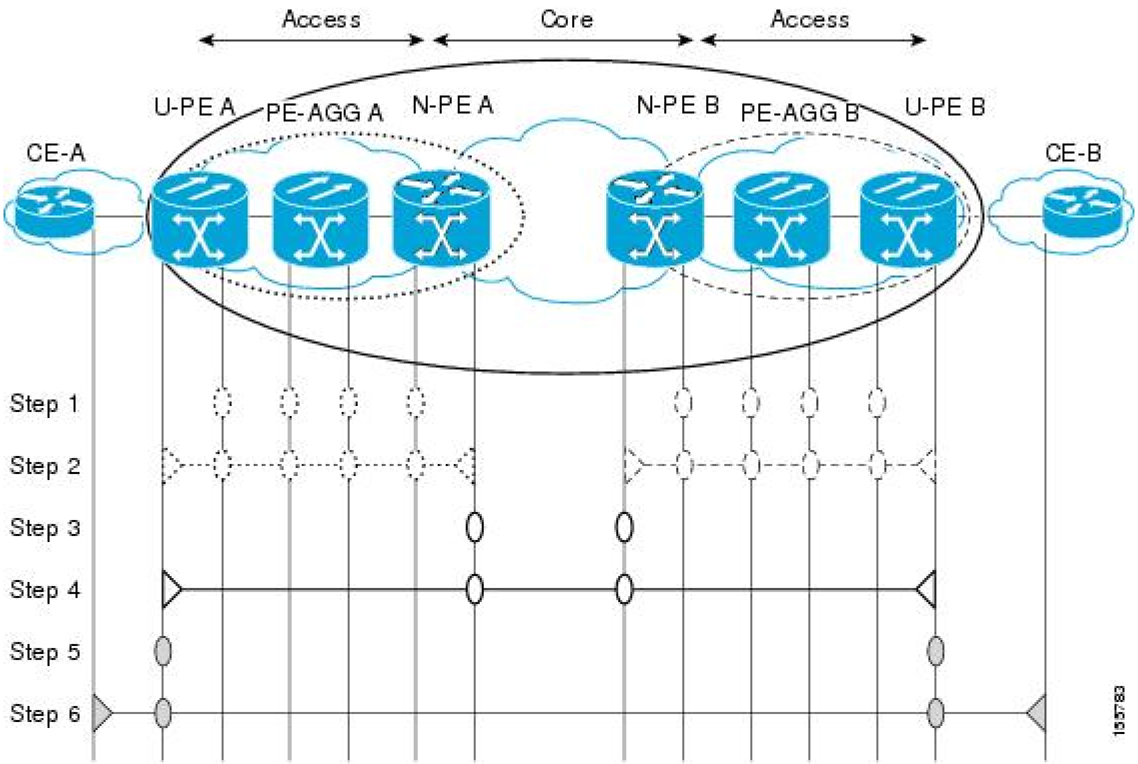
DETAILED STEPS

	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.
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 devices to block CFM frames. • Configure a MIP on every uPE, at the UNI port, in the customer maintenance domain.

	Command or Action	Purpose
		- 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

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 for CE-A

Perform this task to prepare the network for Ethernet CFM.

Before You Begin

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.

	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	**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.

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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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:**	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.

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`
		Returns the CLI to privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router#`	

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.

	Command or Action	Purpose
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.
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:**	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.

	Command or Action	Purpose
Step 11	**interface** *type number* **Example:**	Specifies an interface.
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.

	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 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.

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.

	Command or Action	Purpose
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.
Step 15	**interface** *type number* **Example:**	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.

	Command or Action	Purpose
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.

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.

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.

	Command or Action	Purpose

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.
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:**	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.

	Command or Action	Purpose

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.
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.

	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	**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:**	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:**	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.

	Command or Action	Purpose

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.
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.

	Command or Action	Purpose
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.

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.
Step 15	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet1/2`	Specifies an interface and places the CLI in interface configuration mode.

	Command or Action	Purpose
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.

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.
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.

	Command or Action	Purpose
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
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.

	Command or Action	Purpose
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.

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.

	Command or Action	Purpose

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.
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.

	Command or Action	Purpose
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.

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.

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.

	Command or Action	Purpose
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.

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`	Returns the CLI to global configuration mode.

	Command or Action	Purpose
	Example: `Router(config)#`	
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.
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.

	Command or Action	Purpose
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.

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.

	Command or Action	Purpose
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.
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
Step 11	**interface** *type number* **Example:** `Router(config)# interface gigabitethernet3/1`	Specifies an interface and places the CLI in interface configuration mode.

	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	**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**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• 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.
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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.

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.

	Command or Action	Purpose
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.
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*	Provisions a manual MIP.

	Command or Action	Purpose
	Example: `Router(config-if)# ethernet cfm mip level 4`	• 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.

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**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• 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.
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**]	Enables the transmission of CCMs.

	Command or Action	Purpose
	Example: `Router(config-ecfm-srv)# continuity-check`	
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.

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.

	Command or Action	Purpose
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.
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`	Returns the CLI to global configuration mode.

	Command or Action	Purpose
	Example: `Router(config)#`	
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.
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:**	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.

	Command or Action	Purpose
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:**	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.

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.
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.

	Command or Action	Purpose
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.

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.

	Command or Action	Purpose
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.

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.

	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 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.

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.

	Command or Action	Purpose
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.

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.

	Command or Action	Purpose
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.
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:**	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:**	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.

	Command or Action	Purpose
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.

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.

	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 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.
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.

	Command or Action	Purpose

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.
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.

	Command or Action	Purpose
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.

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`
		Returns the CLI to privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router#`	

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)

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.

	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 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.

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.

	Command or Action	Purpose
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.

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

	Command or Action	Purpose
	Example: `Router(config)#`	
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

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.
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`	Returns the CLI to Ethernet CFM configuration mode.

	Command or Action	Purpose
	Example: `Router(config-ecfm)#`	

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.

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`
		Returns the CLI to privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router#`	

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.

	Command or Action	Purpose
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.
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.

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

	Command or Action	Purpose

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.
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**]	Enables the transmission of CCMs. • The number of CCMs missed before the remote MEP is declared down is set.

	Command or Action	Purpose
	Example: `Router(config-ecfm-srv)# continuity-check loss-threshold 5`	
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.

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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.
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

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 Configuration

```
!
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
 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
 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 Configuration

```
!
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
 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
 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
```

PE-AGG A Configuration

```
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
 ethernet cfm mip level 1 vlan 101   <<<< Manual MIP
!
interface
 ethernet cfm mip level 1    <<<< Manual MIP
```

N-PE A Configuration

```
!
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
 ethernet cfm mip level 1    <<<< manual MIP
!
interface
 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 Configuration

```
!
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
 ethernet cfm mip level 7   <<<< manual MIP
!
interface
 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 Configuration

```
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
 ethernet cfm mip level 2   <<<< manual MIP
!
interface
 ethernet cfm mip level 2   <<<< manual MIP
```

N-PE B Configuration

```
!
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
ethernet cfm mip level 2    <<<< manual MIP
!
interface
 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 Configuration

```
!
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 Configuration

```
!
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
 ethernet cfm mep domain Customer-L7 mpid 701 vlan 101
```

U-PE A Configuration

```
!
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
 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
 ethernet cfm mip level 1 vlan 101   <<<< Manual MIP
```

PE-AGG A Configuration

```
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
 ethernet cfm mip level 1 vlan 101   <<<< Manual MIP
```

```
!
interface
 ethernet cfm mip level 1    <<<< Manual MIP
```

N-PE A Configuration

```
!
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
 ethernet cfm mip level 1    <<<< manual MIP
!
interface
 ethernet cfm mip level 4    <<<< manual MIP
 ethernet cfm mep domain OperatorA mpid 102 vlan 101
```

U-PE B Configuration

```
!
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
 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
 ethernet cfm mip level 2   <<<< manual MIP
```

N-PE B Configuration

```
!
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
ethernet cfm mip level 2      <<<< manual MIP
!
interface
 ethernet cfm mip level 4     <<<< manual MIP
 ethernet cfm mep domain OperatorB mpid 202 vlan 101
```

CE-B Configuration

```
!
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
 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*

Related Topic	Document Title
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	

Feature Name	Releases	Feature Information
		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 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-points local**, **show ethernet cfm maintenance-points**

Feature Name	Releases	Feature Information
		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.

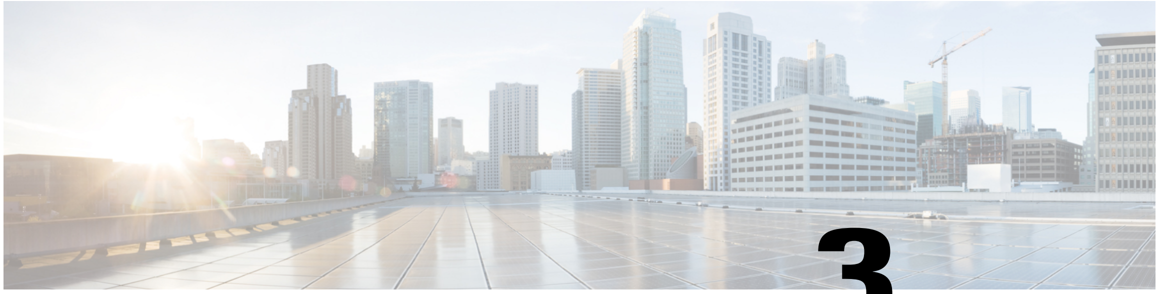
OAM --operations, administration, and maintenance. A term used by several standards bodies to describe protocols and procedures for operating, administrating, and maintaining networks. Examples are ATM OAM and IEEE Std. 802.3ah OAM.

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.



CHAPTER 3

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 caveats and feature information, see Bug Search Tool and 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 module.

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

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

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.

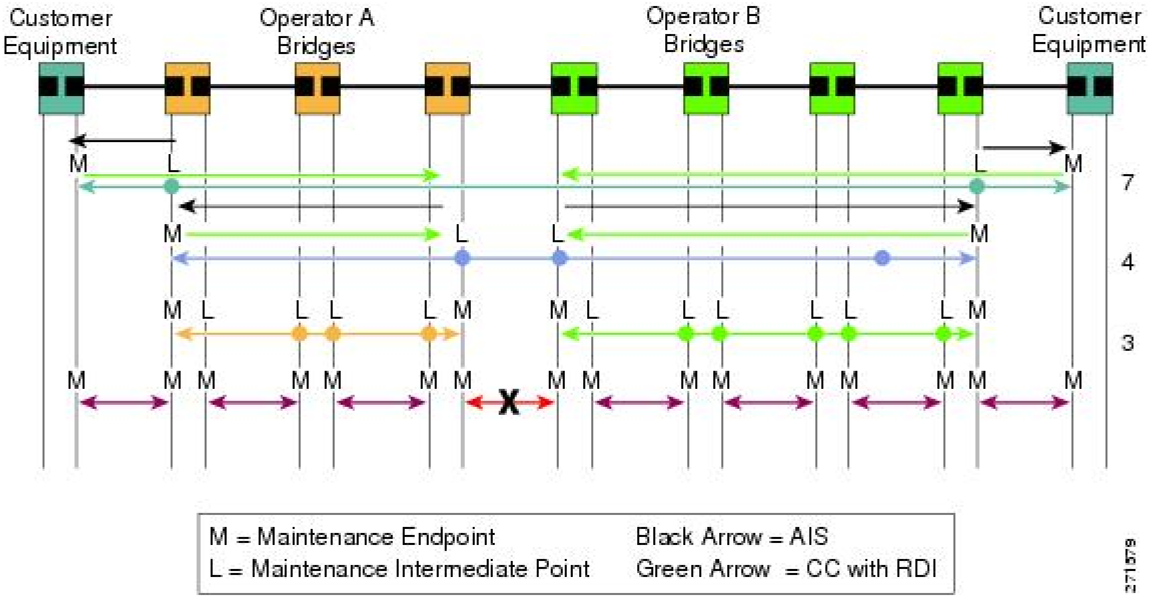
AIS and 802.3ah Interworking

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

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm ais link-status global** **Example:** `Device(config)# ethernet cfm ais link-status global`	Globally enables AIS generation and enters CFM SMEP AIS configuration mode.
Step 4	**disable** **Example:** `Device(config-ais-link-cfm)# disable`	Disables AIS transmission.
Step 5	**exit** **Example:** `Device(config-ais-link-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Device(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:** `Device(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:** `Device(config-ecfm-srv)# no ais`	Disables the AIS function for a specific maintenance association.

Step 9	**end**
	Example:
	`Device(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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Device(config)# ethernet cfm domain PROVIDERDOMAIN level 4`	Defines a CFM maintenance domain at a particular maintenance level and enters 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:** `Device(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:** `Device(config-ecfm-srv)# continuity-check`	Enables the transmission of CCMs.
Step 6	**ais** [**expiry-threshold** *threshold* | **level** *level-id* | **period** *seconds*| **suppress-alarms**] **Example:** `Device(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:** `Device(config-ecfm-srv)# ais level 7`	Enables the AIS function for a specific maintenance association.
Step 8	**exit** **Example:** `Device(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:** `Device(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:** `Device(config-ecfm-srv)# continuity-check`	Enables the transmission of CCMs.
Step 11	**ethernet cfm ais link-status global** **Example:** `Device(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:** `Device(config-ais-link-cfm)# disable`	Disables the generation of AIS frames resulting from a link-status change.
Step 13	**interface** *type number* **Example:** `Device(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:
	`Device(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:** `Device(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:** `Device(config-if)# ethernet cfm ais link-status`	Enables AIS generation from a SMEP.

Step 17	**end**
	Example:
	`Device(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:

```
Device# 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:

```
Device# 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:

```
Device# 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:

```
Device# 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:

```
Device# 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*

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

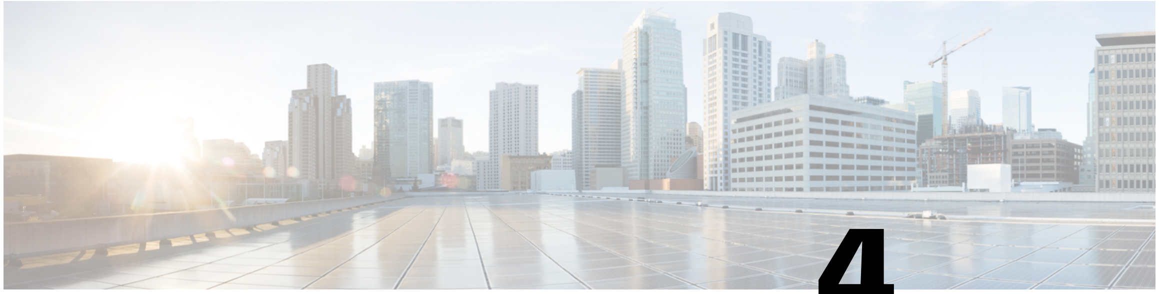
Feature Information for Configuring ITU-T Y.1731 Fault Management Functions

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**.



CHAPTER 4

CFM CCM Extensions to Support the NSN Microwave 1+1 Hot Standby Protocol

The Nokia Siemens Networks (NSN) Microwave 1+1 Hot Standby (HSBY) protocol is a link-protection protocol that extends connectivity fault management (CFM) continuity check messages (CCMs) to enable 1:1 link redundancy in microwave devices. NSN Microwave 1+1 HSBY provides link-protection support for both indoor units (IDUs) and outdoor units (ODUs).

This document describes the extensions to the IEEE 802.1ag CFM component in Cisco IOS software that enable the detection and handling of microwave outdoor unit hardware failures.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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.

Restrictions for CFM CCM Extensions to Support the NSN Microwave 1+1 HSBY Protocol

- NSN Hot Standby supports only the ES+, ES20, and 6700 series line cards on the Cisco 7600 series router.
- To enable link-protection on a maintenance endpoint (MEP), the connectivity fault management (CFM) domain and MEP must adhere to the Nokia Siemens Networks (NSN) configuration requirements.

Information About CFM CCM Extensions to Support the NSN Microwave 1+1 HSBY Protocol

NSN Microwave 1+1 HSBY and CFM Integration

CFM Continuity Check Messages

CFM CCMs are heartbeat messages exchanged periodically between maintenance association endpoints (MEPs). CCMs allow MEPs to discover each other within a maintenance association, and allow maintenance association intermediate points (MIPs) to discover MEPs. CCMs provide a means for detecting connectivity failures in a maintenance domain. CCMs are transmitted frequently enough so that consecutive messages can be lost without causing the information to time out in any of the receiving MEPs.

For detailed information about CFM, MEPs, MIPs, and maintenance associations, see "Configuring IEEE Standard-Compliant Ethernet CFM in a Service Provider Network".

Monitoring Devices and Suspending CFM Traffic

The NSN Microwave 1+1 HSBY Protocol has specified a proprietary time-to-live (TLV) field in CCMs for monitoring active and standby ODUs, and a flag to temporarily suspend CCM monitoring. Identified by an Organizational Unique Identifier (OUI) value of 0x000FBB, the TLV is attached to CCMs as an organization-specific TLV.

An IDU or an ODU may need to temporarily halt transmitting traffic, including CCMs, in circumstances such as a software upgrade or a reload. An IDU or ODUs can set the Suspend CC Monitor flag to signal a temporary pause in CFM traffic if a suspension is needed. Using this flag prevents the other two devices from triggering an unnecessary link-protection action. The Suspend CC Monitor time interval field, in conjunction with the flag, indicates the maximum amount of time the two devices must wait before expecting CCMs to resume from the suspended device.

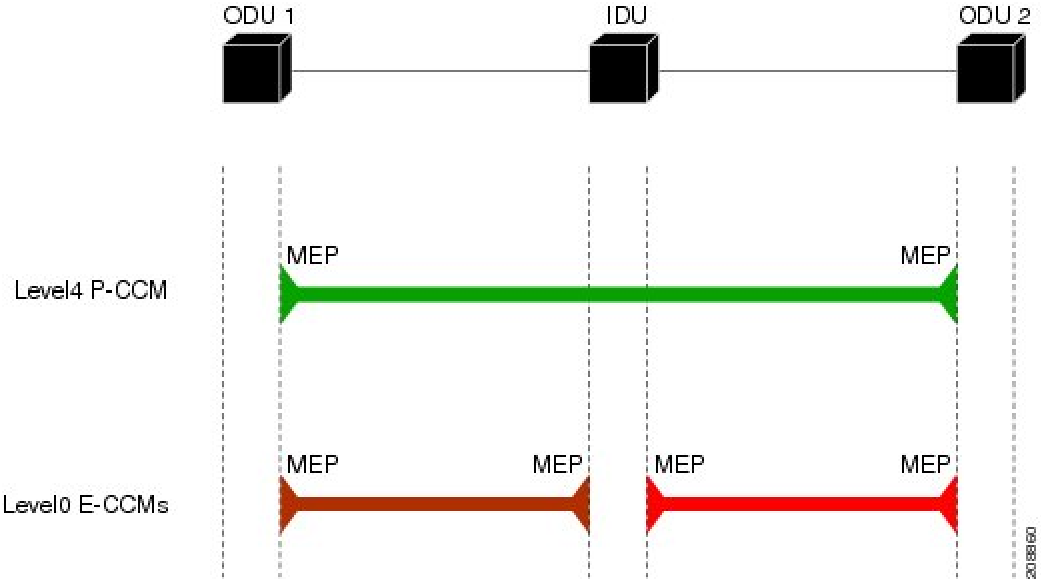
NSN Microwave 1+1 HSBY Protocol Monitoring of Maintenance Associations

The NSN Microwave 1+1 HSBY protocol monitors three maintenance associations. One maintenance association is at Ethernet CFM level 4 and is called the ODU-to-ODU CCM (P-CCM) session, and two maintenance associations are at Ethernet CFM level 0 and are called the IDU-to-ODU CCM (E-CCM) sessions. The IDU is associated with only the two E-CCM sessions and has an outward-facing MEP configured in each session. The IDU is required to pass CFM traffic between the ODUs only in the P-CCM session; no additional monitoring of this maintenance association is needed.

The HSBY configuration shown in the figure below supports four separate traffic flows:

- CFM traffic between the IDU and ODU 1.
- CFM traffic between the IDU and ODU 2.
- CFM traffic between ODU 1 and ODU 2. This traffic passes through the IDU.
- Data traffic between the WAN and ODU 1. This traffic passes through the IDU.

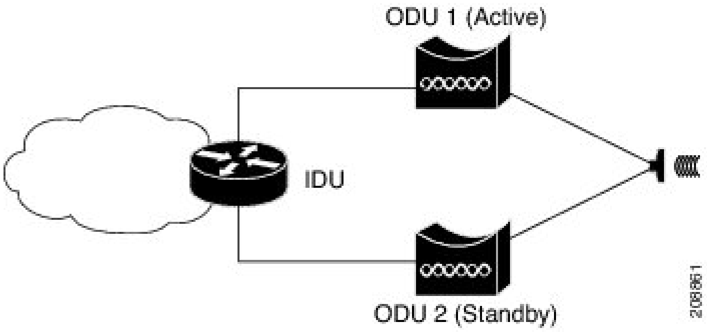
Figure 1: HSBY Protocol and CFM Maintenance Associations



Microwave 1+1 HSBY Configuration

The NSN Microwave 1+1 HSBY link-protection function within the scope of CFM CCM extensions is provided through configuration of a single IDU connected to two ODUs for redundancy. The Cisco IOS device acts as the IDU. At a given time only one ODU is actively handling data traffic, but both the active and standby ODUs are processing and transmitting CFM traffic. The CFM traffic is composed of CCMs with NSN proprietary TLV fields that extend the CCMs’ detection of connectivity failures to IDUs and ODUs. Additionally, these extended CCMs passed between the IDU and ODUs are used to indicate which ODU is active and handling the data traffic. If a failure occurs, the standby ODU assumes the role of the active ODU. The figure below shows a sample physical topology.

Figure 2: HSBY Link Protection Physical Topology



IDU Configuration Values

The HSBY Protocol specifies that some IDU parameters are configurable and others are fixed values. The table below summarizes the permitted values for an IDU using the HSBY Protocol.

Note The same maintenance association (MA) VLAN ID (MA VLAN-ID) can be used for all MAs configured on an IDU.

Table 4: HSBY IDU Configuration Parameters

Parameter	Default Value	Permitted Values
CC Interval	100 milliseconds (ms)	10ms, 100ms, and 1000ms
Domain Level	0	Fixed
Domain Name	Null	Fixed
MA VLAN-ID (E-CCM)	None	1-15
MPID	1	Fixed
Short MA Name	None	0-65535
Suspend Interval	160 seconds	80s, 160s, 240s, and 320s

ODU Configuration Values

The HSBY Protocol specifies that some ODU parameters are configurable and others are fixed values. The table below summarizes the permitted values for an ODU using the HSBY Protocol.

Note By default, an ODU learns the short MA name when it receives the first E-CCM from an IDU.

Table 5: HSBY ODU Configuration Parameters

Parameter	Default Value	Permitted Values
MA VLAN-ID (E-CCM)	None	16-50
MPID	2	Fixed
Short MA Name	Learned	0-65535

How to Configure CFM CCM Extensions to Support the NSN Microwave 1+1 HSBY Protocol

Configuring NSN Microwave 1+1 HSBY Protocol and CFM CCM Extensions

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm global**
4. **link-protection enable**
5. **link-protection group management vlan** *vlan-id*
6. link-protection group *group-number* pccm **vlan** *vlan-id*
7. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
8. **id** {*mac-address domain-number* | **dns** *dns-name* | **null**}
9. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
10. **mep mpid** *mpid*
11. **mep mpid** *mpid*
12. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
13. **exit**
14. **exit**
15. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
16. **id** {*mac-address domain-number* | **dns** *dns-name* | **null**}
17. **service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]]
18. **mep mpid** *mpid*
19. **mep mpid** *mpid*
20. **continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**]
21. **exit**
22. **exit**
23. **interface** *type slot* / *port*
24. **switchport mode** {**access** | **dot1q-tunnel**| **dynamic** {**auto** | **desirable**} | **private-vlan** | **trunk**}
25. **spanning-tree portfast** {**disable** | **trunk**}
26. **ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*}
27. link-protection group *group-number*
28. **exit**
29. **interface** *type slot* / *port*
30. **switchport mode** {**access** | **dot1q-tunnel**| **dynamic** {**auto** | **desirable**} | **private-vlan** | **trunk**}
31. **spanning-tree portfast** {**disable** | **trunk**}
32. **ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*}
33. link-protection group *group-number*
34. **end**
35. **show ethernet cfm maintenance-points remote 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# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm global** **Example:** `Router(config)# ethernet cfm global`	Enables Ethernet CFM globally.
Step 4	**link-protection enable** **Example:** `Router(config)# link-protection enable`	Enables link protection globally on the router.
Step 5	**link-protection group management vlan** *vlan-id* **Example:** `Router(config)# link-protection group management vlan 51`	Defines the management VLAN used for link protection. • The Cisco 7600 series router supports 12 link-protection groups per router.
Step 6	link-protection group *group-number* pccm **vlan** *vlan-id* **Example:** `Router(config)# link-protection group 2 pccm vlan 16`	Specifies an ODU-to-ODU continuity check message (P-CCM) VLAN.
Step 7	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain eccm1 level 0`	Configures the CFM domain for ODU 1 and enters Ethernet CFM configuration mode.

Step 8	**id** {*mac-address domain-number* | **dns** *dns-name* | **null**}
	Example:
	`Router(config-ecfm)# id null`
		Configures a maintenance domain identifier (MDID).

	Command or Action	Purpose
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 1 vlan 14 direction down`	Defines a maintenance association for ODU 1 and enters Ethernet CFM service instance configuration mode.
Step 10	**mep mpid** *mpid* **Example:** `Router(config-ecfm-srv)# mep mpid 1`	Defines the local MEP ID.
Step 11	**mep mpid** *mpid* **Example:** `Router(config-ecfm-srv)# mep mpid 2`	Defines the remote MEP ID.
Step 12	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check interval 100ms`	Enables transmission of continuity check messages (CCMs) within the ODU 1 maintenance association and defines a continuity-check interval.
Step 13	**exit** **Example:** `Router(config-ecfm-srv)# exit`	Exits Ethernet CFM service instance configuration mode.
Step 14	**exit** **Example:** `Router(config-ecfm)# exit`	Exits Ethernet CFM configuration mode.
Step 15	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain eccm2 level 0`	Configures the CFM domain for ODU 2 and enters CFM configuration mode.

Step 16	**id** {*mac-address domain-number* | **dns** *dns-name* | **null**}
	Example:
	`Router(config-ecfm)# id null`
		Configures a maintenance domain identifier (MDID).

	Command or Action	Purpose
Step 17	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Router(config-ecfm)# service 2 vlan 15 direction down`	Defines a maintenance association for ODU 2 and enters Ethernet CFM service configuration mode.
Step 18	**mep mpid** *mpid* **Example:** `Router(config-ecfm-srv)# mep mpid 1`	Defines the local MEP ID.
Step 19	**mep mpid** *mpid* **Example:** `Router(config-ecfm-srv)# mep mpid 2`	Defines the remote MEP ID.
Step 20	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Router(config-ecfm-srv)# continuity-check interval 100ms`	Enables transmission of CCMs within the ODU 2 maintenance association and defines a continuity-check interval.
Step 21	**exit** **Example:** `Router(config-ecfm-srv)# exit`	Exits Ethernet CFM service instance configuration mode.
Step 22	**exit** **Example:** `Router(config-ecfm)# exit`	Exits Ethernet CFM configuration mode.
Step 23	**interface** *type slot / port* **Example:** `Router(config)# interface gigabitethernet 1/1`	Configures the interface to be connected to ODU 1 and enters interface configuration mode.

Step 24	**switchport mode** {**access** | **dot1q-tunnel**| **dynamic** {**auto** | **desirable**} | **private-vlan** | **trunk**}
	Example:
	`Router(config-if)# switchport mode trunk`
		Sets the switching characteristics of the Layer 2-switched interface.

	Command or Action	Purpose

Step 25	**spanning-tree portfast** {**disable** | **trunk**}
	Example:
	`Router(config-if)# spanning-tree portfast trunk`
		Enables PortFast on the interface when it is in trunk mode.

Step 26	**ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep domain eccm1 mpid 1 vlan 14`	Configures a CFM MEP domain for ODU 1.
Step 27	link-protection group *group-number* **Example:** `Router(config-if)# link-protection group 1`	Configures a link-protection group for ODU 2.
Step 28	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 29	**interface** *type slot* / *port* **Example:** `Router(config)# interface GigabitEthernet 3/2`	Configures the interface to be connected to ODU 2 and enters interface configuration mode.

Step 30	**switchport mode** {**access** | **dot1q-tunnel**| **dynamic** {**auto** | **desirable**} | **private-vlan** | **trunk**}
	Example:
	`Router(config-if)# switchport mode trunk`
		Sets the switching characteristics of the Layer 2-switched interface.

Step 31	**spanning-tree portfast** {**disable** | **trunk**}
	Example:
	`Router(config-if)# spanning-tree portfast trunk`
		Enables PortFast on the interface when it is in trunk mode.

Step 32	**ethernet cfm mep domain** *domain-name* **mpid** *mpid* {**port** | **vlan** *vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep domain eccm2 mpid 1 vlan 15`	Configures a CFM MEP domain for ODU 2.

	Command or Action	Purpose
Step 33	link-protection group *group-number* **Example:** `Router(config-if)# link-protection group 1`	Configures a link-protection group for ODU 2.

Step 34	**end**
	Example:
	`Router(config-if)# end`
		Returns the CLI to privileged EXEC mode.

Step 35	**show ethernet cfm maintenance-points remote detail** **Example:** `Router# show ethernet cfm maintenance-points remote detail`	(Optional) Displays remote maintenance endpoints in the continuity check database.

Configuration Examples for CFM CCM Extensions to Support the NSN Microwave 1+1 HSBY Protocol

Example CFM Domain and MEP Configuration

This example is a sample CFM domain and MEP configuration that follows the NSN requirements for monitoring ODUs. The **link-protection** command for configuring NSN-specific parameters is included. CFM configuration parameters for an IDU are shown within angle brackets (<>):

```
link-protection suspend-interval <80s, 160s, 240s, 320s>
link-protection management vlan <51-4094>
link-protection pccm vlan <16-50>
!
ethernet cfm ieee
ethernet cfm global
!
ethernet cfm domain <Domain for ODU1> level 0
 id null
 service number <number> vlan <1-15> direction down
 continuity-check
 continuity-check interval <10, 100, 1000ms>
!
ethernet cfm domain <Domain for ODU2> level 0
 id null
 service number <number> vlan <1-15> direction down
  continuity-check
  continuity-check interval <10, 100, 1000ms>
!
interface GigabitEthernet 0/3
 ethernet cfm mep domain <Domain for ODU1> mpid 1 vlan <1-15>
    link-protection group <group #>
```

```
!
interface GigabitEthernet 0/4
 ethernet cfm mep domain <Domain for ODU2> mpid 1 vlan <1-15>
   link-protection group <group #>
!
```

Example 1+1 HSBY Protocol Configuration

The following example shows a 1+1 HSBY protocol configuration on the Cisco 7600 series router:

```
Router> enable
Router# configure terminal
Router(config)# ethernet cfm global
Router(config)# link-protection enable
Router(config)# link-protection group management vlan 51
Router(config)# link-protection group 2 pccm vlan 16
Router(config)# ethernet cfm domain eccm1 level 0
Router(config-ecfm)# id null
Router(config-ecfm)# service 1 vlan 14 direction down
Router(config-ecfm-srv)# mep mpid 1
Router(config-ecfm-srv)# mep mpid 2
Router(config-ecfm-srv)# continuity-check interval 100ms
Router(config-ecfm-srv)# exit
Router(config-ecfm)# exit
Router(config)# ethernet cfm domain eccm2 level 0
Router(config-ecfm)# id null
Router(config-ecfm)# service 2 vlan 15 direction down
Router(config-ecfm-srv)# mep mpid 1
Router(config-ecfm-srv)# mep mpid 2
Router(config-ecfm-srv)# continuity-check interval 100ms
Router(config-ecfm-srv)# exit
Router(config-ecfm)# exit
Router(config)# interface gigabitethernet 1/1
Router(config-if)# switchport mode trunk
Router(config-if)# spanning-tree portfast trunk
Router(config-if)# ethernet cfm mep domain eccm1 mpid 1 vlan 14
Router(config-if)# link-protection group 1
Router(config-if)# exit
Router(config)# interface GigabitEthernet 3/2
Router(config-if)# switchport mode trunk
Router(config-if)# spanning-tree portfast trunk
Router(config-if)# ethernet cfm mep domain eccm2 mpid 1 vlan 15
Router(config-if)# link-protection group 1
Router(config-if)# end
Router# show ethernet cfm maintenance-points remote detail
```

Additional References for CFM CCM Extensions to Support the NSN Microwave 1+1 Hot Standby Protocol

Related Documents

Related Topic	Document Title
Cisco IOS commands: master list of commands with complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS Master Commands List, All Releases

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*
Configuring IEEE Standard-Compliant Ethernet CFM	"Configuring IEEE Standard-Compliant Ethernet CFM in a Service Provider Network"
Configurations for Carrier Ethernet networks	*Carrier Ethernet Configuration Guide*, Cisco IOS Release 15.1S
Understanding and configuring Microwave 1+1 HSBY on the Cisco MWR 2941 Mobile Wireless Edge Router	“Configuring Ethernet Link Operations, Administration, and Maintenance” chapter of the *Cisco MWR 2941 Mobile Wireless Edge Router Software Configuration Guide, Release 15.0(1)MR*

Standards

Standard	Title
IEEE 802.1ag	*Connectivity Fault Management*

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 CFM CCM Extensions to Support the NSN Microwave 1+1 HSBY Protocol

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 CFM CCM Extensions to Support the NSN Microwave 1+1 HSBY Protocol

Feature Name	Releases	Feature Information
CFM Extension for 1+1 Hot-Standby Support	15.1(2)S	The NSN Microwave 1+1 HSBY protocol is a link-protection protocol that extends CFM CCMs to enable 1:1 link redundancy in microwave devices. NSN Microwave 1+1 HSBY provides link-protection support for both IDUs and ODUs. In Cisco IOS Release 15.1(2)S, this feature was introduced on the Cisco 7600 series router. The following command was introduced or modified: **show ethernet cfm maintenance-points remote detail**.

CHAPTER 5

IEEE-Compliant CFM MIB

The IEEE-compliant CFM MIB (IEEE CFM MIB) provides MIB support for IEEE 802.1ag-compliant connectivity fault management (IEEE CFM) services. The IEEE CFM MIB can be used as a tool to trace paths, verify and manage connectivity, and detect faults in a network.

This document describes the IEEE CFM MIB and the IEEE CFM services that it supports.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 the IEEE-Compliant CFM MIB

- The CFM 8.1ag module must be present in the software image.

Restrictions for the IEEE-Compliant CFM MIB

- The IEEE-compliant CFM MIB does not support SET operations.

- The IEEE-compliant CFM MIB does not support the capability to create rows.
- Some devices do not support the CFM MIB on bridge domains and IETF. See to the documentation for the device.

Information About the IEEE-Compliant CFM MIB

IEEE CFM MIB Implementation

The IEEE CFM MIB is compliant with the IEEE 802.1ap standard, which defines the IEEE CFM MIB as two modules: IEEE8021-CFM-MIB and IEEE8021-CFM-V2-MIB. The IEEE CFM MIB feature implements both modules.

The IEEE 802.1ag standard provides capabilities for detecting and isolating connectivity failures in a network. Network operators need network management tools to trace paths, verify and manage connectivity, and detect faults in a network. The IEEE CFM MIB has tables and objects that can be created, written, and read by network administrators. Additionally, an administrator may provide limited access to another provider, who can become the administrator of that table or object. Some tables and objects (for example, the CFM stack managed object or the default maintenance domain level object) can be used only by the owner of the network or bridge.

IEEE CFM Services Supported by the IEEE CFM MIB

The table below shows the IEEE CFM services that the IEEE CFM MIB supports and the MIB modules associated with the service.

Table 7: IEEE CFM Services Supported by the IEEE-compliant CFM MIB

CFM Service	Required IF-MIB Support	Associated Service MIB Modules
CFMoVLANs	IF-MIB support for interfaces	--
CFMoEVC with BD	IF-MIB support for Ethernet flow points (EFPs) forwarding defined as bridge domain	• CISCO-EVC-MIB • CISCO-BRIDGE-DOMAIN-MIB
CFMoEVC with XC	IF-MIB support for EFPs forwarding defined as cross connect	• CISCO-EVC-MIB • CISCO-IETF-PW-MIB

Tables in the IEEE CFM MIB

Eleven tables and one set of alarms are in the IEEE CFM MIB and are listed by module in the following sections:

IEEE8021-CFM-MIB

Six tables and the set of alarms are in the IEEE8021-CFM-MIB module. The tables are:

- Linktrace Reply Table (dot1agCfmLtrTable)--Extends the maintenance endpoint (MEP) table and contains a list of linktrace replies received by a specific MEP in response to a linktrace message.
- Maintenance Association (MA) Network Table (dot1agCfmMaNetTable)--Lists maintenance associations; each row in the table represents a maintenance association. This part of the MA table is constant across all bridges in a maintenance domain or across all components of a single bridge.
- Maintenance Domain Table (dot1agCfmMdTable)--Lists maintenance domains; each row in the table represents a different maintenance domain.
- MAMEP Table (dot1agCfmMaMepListTable)--Lists a table entry for known MEPs for an MA.
- MEP Database Table (dot1agCfmMepDbTable)--Extends the MEP table and is a database of information received about other MEPs in the maintenance domain.
- MEP Table (dot1agCfmMepTable)--Lists MEPs; each row in the table represents a different MEP.

A fault alarm (notification or trap) is sent to the management entity when a defect condition is detected. The object identifier (OID) of the MEP that detected the defect condition is sent as part of the alarm.

Fault alarms are assigned priorities, which perform the following functions:

- Define that a period of time should elapse with the defect condition present before a fault alarm is sent. The default is 2.5 seconds.
- Define that alarms are not sent after a time period has elapsed in which no alarms occurred. The default is 10 seconds.
- Define an alarm priority from 5 (highest) to 1 (lowest) that controls which failures trigger fault alarms.

Defect Descriptions and Associated Fault Alarm Priorities

The table below shows the defects listed in order of their associated priorities.

Table 8: Defect Descriptions and Associated Fault Alarm Priorities

Defect	Priority	Description
DefXconCCM	5 (highest)	One or more cross-connect continuity check messages (CCMs) has been received, and 3.5 times at least one of those CCMs' transmission interval has not yet expired: CrossConnect Error.
DefErrorCCM	4	One or more invalid CCMs has been received, and 3.5 times the CCMs' transmission interval has not yet expired: CrossCheck Error--Unknown MEP, Config Error, Loop Error.

Defect	Priority	Description
DefRemoteCCM	3	At least one of the remote MEP state machines is not receiving valid CCMs from its remote MEP: CrossCheck Error--MEP Missing.
DefMACstatus	2	One or more of the remote MEPs is reporting a failure in its Port Status Type-Length-Value (TLV) or Interface Status TLV: MEP Down.
DefRDICCM	1 (lowest)	At least one of the Remote MEP state machines is receiving valid CCMs from its remote MEP that has the Remote Defect Indication (RDI) bit set.

IEEE8021-CFM-V2-MIB

The following five tables are in the IEEE8021-CFM-V2-MIB module:

- CFM Configuration Error List Table (ieee8021CfmConfigErrorListTable)--Provides lists of interfaces and VIDs that are incorrectly configured.
- CFM Default MD Level Table (ieee8021CfmDefaultMdTable)--For each bridge component, controls MIP Half Function (MHF) creation for VIDs that are not attached to a specific maintenance association managed object and Sender ID TLV transmission by those MHFs.
- CFM Maintenance Association Component Table (ieee8021CfmMaCompTable)--Lists maintenance associations. Each row in the table represents a maintenance association. This part of the MA table is variable across bridges in a maintenance domain or across components of a single bridge.
- CFM Stack Table (ieee8021CfmStackTable)--Allows retrieval of information about maintenance points configured on any interface. There is one stack table per bridge.
- CFM VLAN Table (ieee8021CfmVlanTable)--Defines the association of primary VIDs with VLANs. Each VID that is not the primary VID and each VID that belongs to a VLAN associated with more than one VID has an entry in this table. VLANs associated with a single VID should not have an entry in this table.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands: master list of commands with complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS Master Command 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*
Configuring Carrier Ethernet	*Carrier Ethernet Configuration Guide*

Standards

Standard	Title
IEEE 802.1ag-2007	*IEEE Standard for Local and metropolitan area networks--Virtual Bridged Local Area Networks*
IEEE 802.1ap	*802.1ap - Management Information Base (MIB) definitions for VLAN Bridges*

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 the IEEE-Compliant CFM MIB

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 the IEEE-Compliant CFM MIB

Feature Name	Releases	Feature Information
IEEE-Compliant CFM MIB	Cisco IOS XE Release 3.8S	The IEEE-compliant CFM MIB provides MIB support for IEEE CFM services. The IEEE-compliant CFM MIB can be used as a tool to trace paths, verify and manage connectivity, and detect faults in a network. The following commands were modified: **ethernet cfm alarm**, **snmp-server enable traps ethernet cfm alarm**, **snmp-server host**.

CHAPTER 6

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.

Note As an alternative, CFM can be configured over an Ethernet flow point (EFP) interface by using the cross connect functionality. For more information about this alternative, see Configuring the CFM over EFP Interface with Cross Connect Feature.

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.

Restrictions for Configuring Ethernet CFM in a Service Provider Network

- 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.
- 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.
- CFM configuration is not supported on an EtherChannel in FastEthernet Channel (FEC) mode.

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.

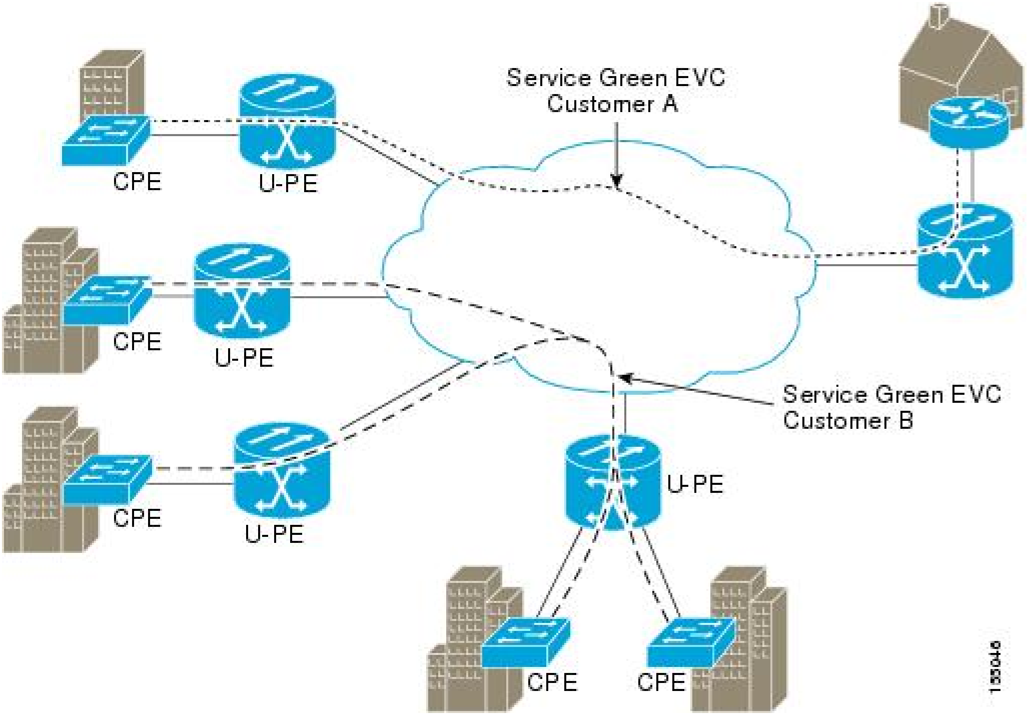
Troubleshooting carrier networks offering Ethernet Layer 2 services is challenging. Customers contract with service providers for end-to-end Ethernet service and service providers may subcontract with operators to provide equipment and networks. Compared to enterprise networks, where Ethernet traditionally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

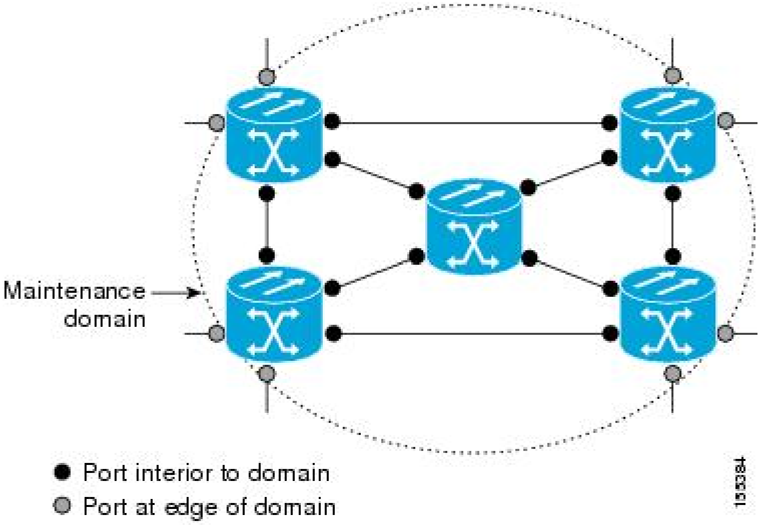
Customer Service Instance

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.

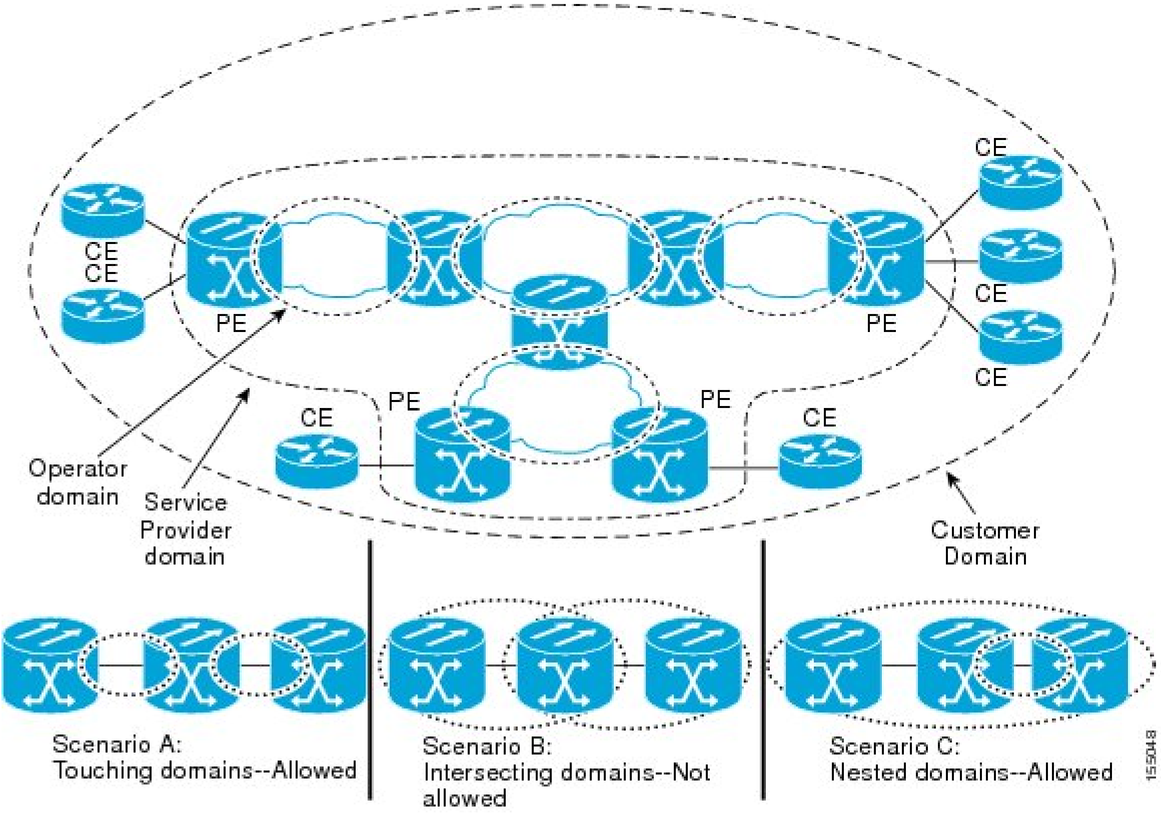


A unique maintenance level in the range of 0 to 7 is assigned to each domain by a network administrator. Levels and domain names are useful for defining the hierarchical relationship that exists among domains. The hierarchical relationship of domains parallels the structure of customer, service provider,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


CE
CE
CE
CE
CE
CE
CE
CE
PE
PE
PE
PE
Operator domain
Service Provider domain
Customer Domain
Scenario A: Touching domains--Allowed
Scenario B: Intersecting domains--Not allowed
Scenario C: Nested domains--Allowed
155048

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

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 Connectivity Fault Management (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 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 Port Channels

Outward facing means that the MEP communicates through the wire. Outward facing MEPs can be configured on port channels (using cross connect functionality). A MIP configuration at a level higher than the level of the outward facing MEP is not required.

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.

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.
- If the port on which the outward MEP is configured is blocked by the Spanning-Tree Protocol, the MEP can still transmit and receive CFM messages via the wire.

Maintenance Intermediate Points

MIPs have the following characteristics:

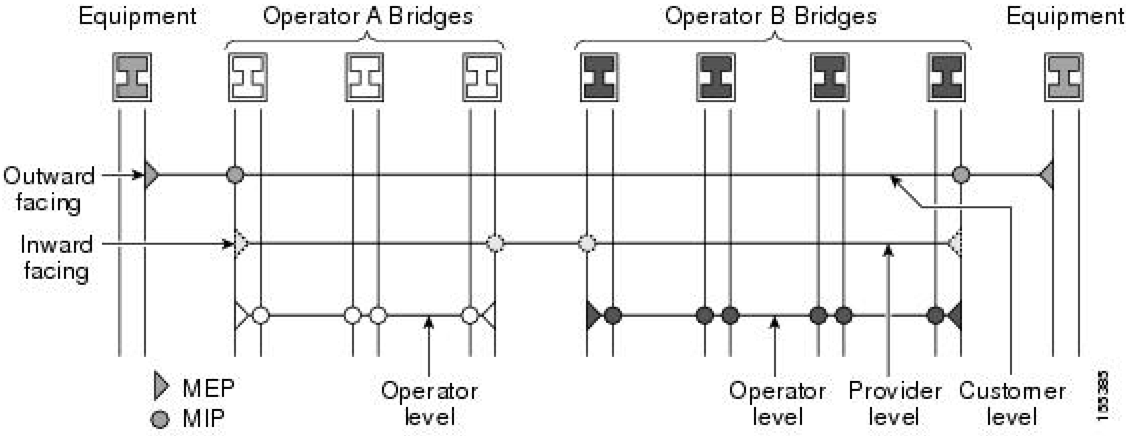
- Per maintenance domain (level) and for all S-VLANs enabled or allowed on a port.

- 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 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

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 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

Connectivity Fault Management (CFM) over bridge domains allows untagged CFM packets to be associated with a maintenance end point (MEP). An incoming untagged customer CFM packet has an EtherType of CFM and is mapped to an Ethernet virtual circuit (EVC) or bridge domain based on the encapsulation configured on the Ethernet flow point (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 an ATM virtual circuit. 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.

HA Features Supported by CFM

In access and service provider networks using Ethernet technology, High Availability (H)A 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 Switch Processor (RSP).

Note A hot standby Route Switch Processor (RSP) has the same software image as the active RSP and supports synchronization of protocol and application state information between RSPs for supported features and protocols.

End-to-end connectivity status is maintained on the customer edge (CE), provider edge (PE), and access aggregation PE (uPE) network nodes based on information received by protocols such as Connectivity Fault Management (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 infrastructure provides various component application program interfaces (APIs) that help to maintain a hot standby RSP. Metro Ethernet HA clients 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 software image upgrades, allowing for faster upgrades.
- 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.
- Reduced operating costs due to outages while delivering higher 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

A standalone Connectivity Fault Management (CFM) implementation does not have explicit high availability (HA) requirements. When CFM is implemented on a customer edge (CE) or provider edge (PE), CFM must maintain the Ethernet virtual circuit (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; consequently HA requirements vary for CE and PE.

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.

The PE has EVC configuration and associated customer VLAN information and derives the EVC state and remote UNI from CFM.

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 Stateful Switchover (SSO) and Nonstop Forwarding (NSF) are both supported in Ethernet Connectivity Fault Management (CFM) and are automatically enabled. A switchover from an active to a standby Route Switch Processor (RSP) occurs when the active RSP 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 RSP switchover.

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

ISSU Support in CFM 802.1ag 1.0d

In Service Upgrades (ISSUs) allow you to perform a Cisco software upgrade or downgrade without disrupting packet flow. Connectivity Fault Management (CFM) performs a bulk update and a runtime update of the continuity check database to the standby Route Switch Processor (RSP), 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 RSP to standby RSP 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 “Performing an In Service Software Upgrade ” module of the *High Availability Configuration Guide*.

How to Set Up Ethernet CFM in a Service Provider Network

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.

Before You Begin

- 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

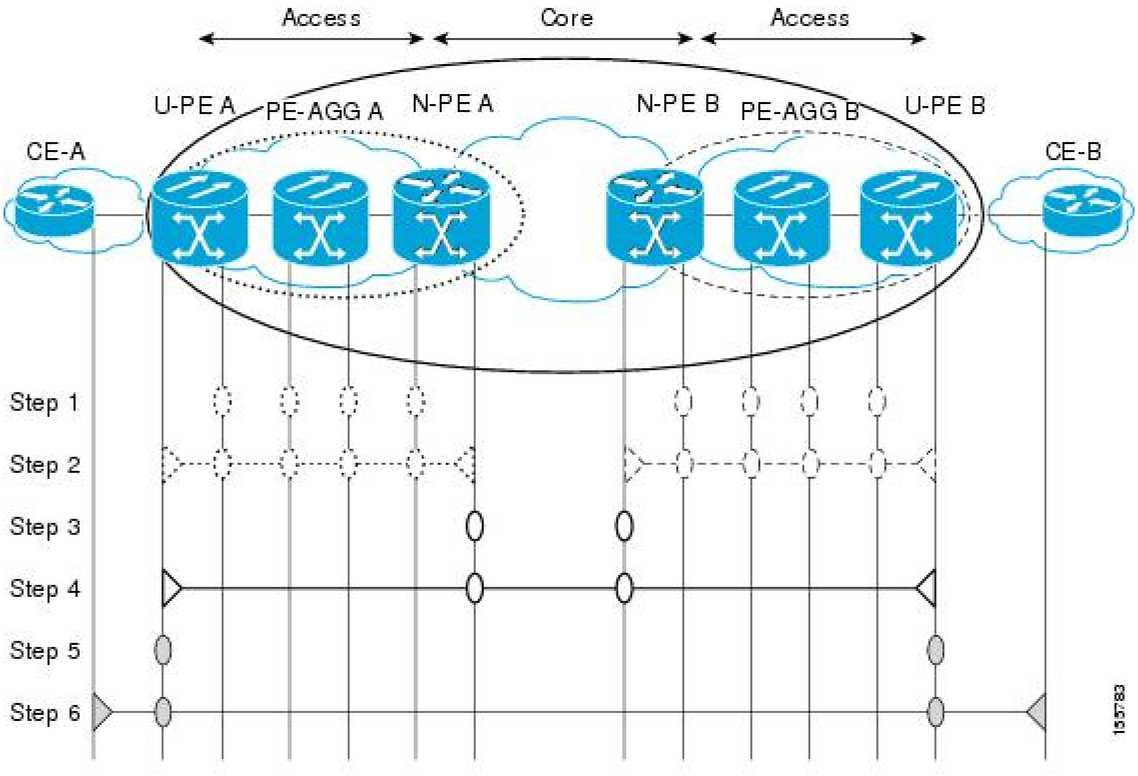
	Command or Action	Purpose
Step 1	Determine operator level MIPs.	Follow these steps: • Starting at lowest operator level domain, assign a MIP at every interface internal to the operator network to be visible to CFM.

	Command or Action	Purpose
		• 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.
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 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.

	Command or Action	Purpose
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

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

Provisioning the Network on the CE-A

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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 9	**exit** **Example:** `Device(config-ecfm)# exit`	Returns to global configuration mode.
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 11	**ethernet cfm traceroute cache** **Example:** `Device(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Device(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:** `Device(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	**snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Device(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 15	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown** | **mep-missing** | **service-up**] **Example:** `Device(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 16	**end**
	Example:
	`Device(config)# end`
		Returns 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. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
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. **service instance** *id* **ethernet** [*evc-name*]
16. **encapsulation** *encapsulation-type*
17. **bridge-domain** *bridge-id*
18. **cfm mip level** { *level* }
19. **exit**
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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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 9	**exit** **Example:** `Device(config-ecfm)# exit`	Returns to global configuration mode.
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 11	**ethernet cfm traceroute cache** **Example:** `Device(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.

	Command or Action	Purpose
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Device(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:** `Device(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:**	Specifies an interface and enters interface configuration mode.
Step 15	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 17	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.

Step 18	**cfm mip level** { *level* }
	Example:
	`Device(config-if-srv)#cfm mip level 4`
		Creates a MIP and sets the maintenance level number.

Step 19	**exit** **Example:** `Device(config-if-srv)# exit`	Returns to interface configuration mode.
Step 20	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 21	**snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Device(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:** `Device(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 23	**end**
	Example:
	`Device(config)# end`
		Returns 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. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
11. **interface** *type number*
12. **service instance** *id* **ethernet** [*evc-name*]
13. **encapsulation** *encapsulation-type*
14. **bridge-domain** *bridge-id*
15. **cfm mip level** *level*
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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 9	**exit** **Example:** `Device(config-ecfm)# exit`	Returns the CLI to global configuration mode.
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 11	**interface** *type* *number* **Example:**	Specifies an interface and enters interface configuration mode.
Step 12	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 13	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 14	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 15	**cfm mip level** *level* **Example:** `Device(config-if-srv)#cfm mip level 4`	Creates a MIP and sets the maintenance level number.

Step 16	**end**
	Example:
	`Device(config-if)# end`
		Returns 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. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **ethernet cfm global**
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. **service instance** *id* **ethernet** [*evc-name*]
14. **encapsulation** *encapsulation-type*
15. **bridge-domain** *bridge-id*
16. **cfm mip level** *level*
17. **exit**
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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 9	**ethernet cfm traceroute cache** **Example:** `Device(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 10	**ethernet cfm traceroute cache size** *entries* **Example:** `Device(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:** `Device(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:**	Specifies an interface and enters interface configuration mode.
Step 13	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 14	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 15	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 16	**cfm mip level** *level* **Example:** `Device(config-if-srv)#cfm mip level 4`	Creates a MIP and sets the maintenance level number.
Step 17	**exit** **Example:** `Device(config-if-srv)# exit`	Returns to interface configuration mode.
Step 18	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.
Step 19	**snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Device(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:** `Device(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 21	**end**
	Example:
	`Device(config)# end`
		Returns 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*
4. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
11. **ethernet cfm traceroute cache**
12. **ethernet cfm traceroute cache size** *entries*
13. **ethernet cfm traceroute cache hold-time** *minutes*
14. **snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**]
15. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown** | **mep-missing** | **service-up**]
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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 9	**exit** **Example:** `Device(config-ecfm)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 11	**ethernet cfm traceroute cache** **Example:** `Device(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Device(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:** `Device(config)# ethernet cfm traceroute cache hold-time 60`	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	**snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Device(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 15	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown** | **mep-missing** | **service-up**] **Example:** `Device(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 16	**end**
	Example:
	`Device(config)# end#`
		Returns 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. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
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. **service instance** *id* **ethernet** [*evc-name*]
16. **encapsulation** *encapsulation-type*
17. **bridge-domain** *bridge-id*
18. **cfm mip level** *level*
19. **exit**
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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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 9	**exit** **Example:** `Device(config-ecfm)# exit`	Returns to global configuration mode.
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 11	**ethernet cfm traceroute cache** **Example:** `Device(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.

	Command or Action	Purpose
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Device(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:** `Device(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:**	Specifies an interface and enters interface configuration mode.
Step 15	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 17	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 18	**cfm mip level** *level* **Example:** `Device(config-if-srv)#cfm mip level 4`	Creates a MIP and sets the maintenance level number.
Step 19	**exit** **Example:** `Device(config-if-srv)# exit`	Returns to interface configuration mode.
Step 20	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 21	**snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Device(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:** `Device(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 23	**end**
	Example:
	`Device(config)# end`
		Returns 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. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
11. **interface** *type number*
12. **service instance** *id* **ethernet** [*evc-name*]
13. **encapsulation** *encapsulation-type*
14. **bridge-domain** *bridge-id*
15. **cfm mip level** *level*
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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:** `Device(config-ecfm)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 11	**interface** *type number* **Example:**	Specifies an interface and enters interface configuration mode.
Step 12	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 13	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 14	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 15	**cfm mip level** *level* **Example:** `Device(config-if-srv)#cfm mip level 4`	Creates a MIP and sets the maintenance level number.

Step 16	**end**
	Example:
	`Device(config-if-srv)# end`
		Returns to privileged EXEC mode.

Provisioning the Network on the N-PE B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
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. **service instance** *id* **ethernet** [*evc-name*]
16. **encapsulation** *encapsulation-type*
17. **bridge-domain** *bridge-id*
18. **cfm mip level** *level*
19. **exit**
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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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 9	**exit** **Example:** `Device(config-ecfm)# exit`	Returns to global configuration mode.
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 11	**ethernet cfm traceroute cache** **Example:** `Device(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.

	Command or Action	Purpose
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Device(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:** `Device(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:**	Specifies an interface and enters interface configuration mode.
Step 15	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 17	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 18	**cfm mip level** *level* **Example:** `Device(config-if-srv)#cfm mip level 4`	Creates a MIP and sets the maintenance level number.
Step 19	**exit** **Example:** `Device(config-if-srv)# exit`	Returns to interface configuration mode.
Step 20	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 21	**snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Device(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:** `Device(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 23	**end**
	Example:
	`Device(config)# end`
		Returns 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*
4. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
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. **service instance** *id* **ethernet** [*evc-name*]
16. **encapsulation** *encapsulation-type*
17. **bridge-domain** *bridge-id*
18. **cfm mep domain** *domain-name* **mpid** *id*
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(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	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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 9	**exit** **Example:** `Device(config-ecfm)# exit`	Returns to global configuration mode.
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 11	**ethernet cfm traceroute cache** **Example:** `Device(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Device(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.

	Command or Action	Purpose
Step 13	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Device(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:**	Specifies an interface and enters interface configuration mode.
Step 15	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 17	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 18	**cfm mep domain** *domain-name* **mpid** *id* **Example:** `Device(config-if-srv)# cfm mep domain L4 mpid 4001`	Configures the MEP domain and the ID.

Step 19	**end**
	Example:
	`Device(config-if-srv)# end`
		Returns 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. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
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. **service instance** *id* **ethernet** [*evc-name*]
16. **encapsulation** *encapsulation-type*
17. **bridge-domain** *bridge-id*
18. **cfm mep domain** *domain-name* **mpid** *id*
19. **exit**
20. **exit**
21. **interface** *type number*
22. **service instance** *id* **ethernet** [*evc-name*]
23. **encapsulation** *encapsulation-type*
24. **bridge-domain** *bridge-id*
25. **cfm mip level** *level*
26. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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 9	**exit** **Example:** `Device(config-ecfm)# exit`	Returns to global configuration mode.
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 11	**ethernet cfm traceroute cache** **Example:** `Device(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Device(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:** `Device(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:**	Specifies an interface and enters interface configuration mode.
Step 15	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 17	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 18	**cfm mep domain** *domain-name* **mpid** *id* **Example:** `Device(config-if-srv)# cfm mep domain L4 mpid 4001`	Configures the MEP domain and the ID.
Step 19	**exit** **Example:** `Device(config-if-srv)# exit`	Returns to interface configuration mode.
Step 20	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 21	**interface** *type number* **Example:**	Specifies an interface and enters interface configuration mode.
Step 22	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 23	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 24	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 25	**cfm mip level** *level* **Example:** `Device(config-if-srv)#cfm mip level 4`	Creates a MIP and sets the maintenance level number.

Step 26	**end**
	Example:
	`Device(config-if-srv)# end`
		Returns 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. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
11. **interface** *type number*
12. **service instance** *id* **ethernet** [*evc-name*]
13. **encapsulation** *encapsulation-type*
14. **bridge-domain** *bridge-id*
15. **cfm mip level** *level*
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.

	Command or Action	Purpose
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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:** `Device(config-ecfm)# exit`	Returns to global configuration mode.
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 11	**interface** *type number* **Example:**	Specifies an interface and enters interface configuration mode.
Step 12	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 13	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 14	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.

	Command or Action	Purpose
Step 15	**cfm mip level** *level* **Example:** `Device(config-if-srv)#cfm mip level 4`	Creates a MIP and sets the maintenance level number.

Step 16	**end**
	Example:
	`Device(config-if-srv)# end`
		Returns 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. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
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. **service instance** *id* **ethernet** [*evc-name*]
16. **encapsulation** *encapsulation-type*
17. **bridge-domain** *bridge-id*
18. **cfm mip level** *level*
19. **exit**
20. **exit**
21. **interface** *type number*
22. **service instance** *id* **ethernet** [*evc-name*]
23. **encapsulation** *encapsulation-type*
24. **bridge-domain** *bridge-id*
25. **cfm mep domain** *domain-name* **mpid** *id*
26. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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 9	**exit** **Example:** `Device(config-ecfm)# exit`	Returns to global configuration mode.
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 11	**ethernet cfm traceroute cache** **Example:** `Device(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Device(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:** `Device(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:**	Specifies an interface and enters interface configuration mode.
Step 15	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 17	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 18	**cfm mip level** *level* **Example:** `Device(config-if-srv)#cfm mip level 4`	Creates a MIP and sets the maintenance level number.
Step 19	**exit** **Example:** `Device(config-if-srv)# exit`	Returns to interface configuration mode.
Step 20	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 21	**interface** *type number* **Example:**	Specifies an interface.
Step 22	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 23	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 24	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 25	**cfm mep domain** *domain-name* **mpid** *id* **Example:** `Device(config-if-srv)# cfm mep domain L4 mpid 4001`	Configures the MEP domain and the ID.

Step 26	**end**
	Example:
	`Device(config-if-srv)# end`
		Returns 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*
4. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
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. **service instance** *id* **ethernet** [*evc-name*]
16. **encapsulation** *encapsulation-type*
17. **bridge-domain** *bridge-id*
18. **cfm mep domain** *domain-name* **mpid** *id*
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(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	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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 9	**exit** **Example:** `Device(config-ecfm)# exit`	Returns to global configuration mode.
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 11	**ethernet cfm traceroute cache** **Example:** `Device(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Device(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table.

	Command or Action	Purpose
Step 13	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Device(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:**	Specifies an interface and enters interface configuration mode.
Step 15	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 17	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 18	**cfm mep domain** *domain-name* **mpid** *id* **Example:** `Device(config-if-srv)# cfm mep domain L4 mpid 4001`	Configures the MEP domain and the ID.

Step 19	**end**
	Example:
	`Device(config-if-srv)# end`
		Returns 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. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
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. **service instance** *id* **ethernet** [*evc-name*]
16. **encapsulation** *encapsulation-type*
17. **bridge-domain** *bridge-id*
18. **cfm mip level** *level*
19. **exit**
20. **exit**
21. **interface** *type number*
22. **service instance** *id* **ethernet** [*evc-name*]
23. **encapsulation** *encapsulation-type*
24. **bridge-domain** *bridge-id*
25. **cfm mep domain** *domain-name* **mpid** *id*
26. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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 9	**exit** **Example:** `Device(config-ecfm)# exit`	Returns to global configuration mode.
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 11	**ethernet cfm traceroute cache** **Example:** `Device(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Device(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:** `Device(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:**	Specifies an interface and enters interface configuration mode.
Step 15	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 17	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 18	**cfm mip level** *level* **Example:** `Device(config-if-srv)#cfm mip level 4`	Creates a MIP and sets the maintenance level number.
Step 19	**exit** **Example:** `Device(config-if-srv)# exit`	Returns to interface configuration mode.
Step 20	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 21	**interface** *type number* **Example:**	Specifies an interface and enters interface configuration mode.
Step 22	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 23	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 24	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 25	**cfm mep domain** *domain-name* **mpid** *id* **Example:** `Device(config-if-srv)# cfm mep domain L4 mpid 4001`	Configures the MEP domain and the ID.

Step 26	**end**
	Example:
	`Device(config-if-srv)# end`
		Returns 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. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
11. **interface** *type number*
12. **service instance** *id* **ethernet** [*evc-name*]
13. **encapsulation** *encapsulation-type*
14. **bridge-domain** *bridge-id*
15. **cfm mip level** *level*
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.

	Command or Action	Purpose
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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:** `Device(config-ecfm)# exit`	Returns to global configuration mode.
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 11	**interface** *type number* **Example:**	Specifies an interface and enters interface configuration mode.
Step 12	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 13	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 14	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.

	Command or Action	Purpose
Step 15	**cfm mip level** *level* **Example:** `Device(config-if-srv)#cfm mip level 4`	Creates a MIP and sets the maintenance level number.

Step 16	**end**
	Example:
	`Device(config-if-srv)# end`
		Returns 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. **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
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. **service instance** *id* **ethernet** [*evc-name*]
16. **encapsulation** *encapsulation-type*
17. **bridge-domain** *bridge-id*
18. **cfm mip level** *level*
19. **exit**
20. **exit**
21. **interface** *type number*
22. **service instance** *id* **ethernet** [*evc-name*]
23. **encapsulation** *encapsulation-type*
24. **bridge-domain** *bridge-id*
25. **cfm mep domain** *domain-name* **mpid** *id*
26. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Configures the transmission of continuity check messages (CCMs).
Step 6	**continuity-check** [**interval** *cc-interval*] **Example:** `Device(config-ecfm-srv)# continuity-check interval 10s`	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(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 9	**exit** **Example:** `Device(config-ecfm)# exit`	Returns to global configuration mode.
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 11	**ethernet cfm traceroute cache** **Example:** `Device(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Device(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:** `Device(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:**	Specifies an interface and enters interface configuration mode.
Step 15	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 17	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 18	**cfm mip level** *level* **Example:** `Device(config-if-srv)#cfm mip level 4`	Creates a MIP and sets the maintenance level number.
Step 19	**exit** **Example:** `Device(config-if-srv)# exit`	Returns to interface configuration mode.
Step 20	**exit** **Example:** `Device(config-if)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 21	**interface** *type number* **Example:**	Specifies an interface.
Step 22	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Device(config-if)# service instance 333 ethernet evc1`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 23	**encapsulation** *encapsulation-type* **Example:**	Sets the encapsulation method used by the interface.
Step 24	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 25	**cfm mep domain** *domain-name* **mpid** *id* **Example:** `Device(config-if-srv)# cfm mep domain L4 mpid 4001`	Configures the MEP domain and the ID.

Step 26	**end**
	Example:
	`Device(config-if-srv)# end`
		Returns 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

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(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:** `Device(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:** `Device(config-ether-cfm)# exit#`	Returns to global configuration mode.
Step 6	**ethernet cfm mep crosscheck start-delay** *delay* **Example:** `Device(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

	Command or Action	Purpose
Step 7	**exit** **Example:** `Device(config)# exit`	Returns 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:** `Device# 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(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:** `Device(config-ether-cfm)# mep crosscheck mpid 401 vlan 100`	Statically defines a remote MEP on a specified VLAN within the domain.
Step 5	**exit** **Example:** `Device(config-ether-cfm)# exit`	Returns to global configuration mode.
Step 6	**ethernet cfm mep crosscheck start-delay** *delay* **Example:** `Device(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:** `Device(config)# exit`	Returns 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:** `Device# 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Device(config)# ethernet cfm domain Customer level 7 direction outward`	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:** `Device(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:** `Device(config-ether-cfm)# exit`	Returns to global configuration mode.
Step 6	**ethernet cfm mep crosscheck start-delay** *delay* **Example:** `Device(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:** `Device(config)# exit`	Returns 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:** `Device# 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Device(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:** `Device(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:** `Device(config-ether-cfm)# exit`	Returns to global configuration mode.
Step 6	**ethernet cfm mep crosscheck start-delay** *delay* **Example:** `Device(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:** `Device(config)# exit`	Returns 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:** `Device# 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **direction outward** **Example:** `Device(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:** `Device(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:** `Device(config-ether-cfm)# exit`	Returns to global configuration mode.
Step 6	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(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:** `Device(config-ether-cfm)# exit`	Returns to global configuration mode.
Step 8	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain PROVIDER level 4`	Defines a CFM maintenance domain at a particular level and enters Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 9	**service** *csi-id* **evc** *evc-name* **Example:** `Device(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:** `Device(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:** `Device(config-ether-cfm)# exit`	Returns to global configuration mode.
Step 12	**ethernet evc** *evc-name* **Example:** `Device(config)# ethernet evc evc_100`	Defines an EVC and enters EVC configuration mode.
Step 13	**exit** **Example:** `Device(config-evc)# exit`	Returns to global configuration mode.
Step 14	**interface** *type number* **Example:**	Specifies an interface and enters interface configuration mode.
Step 15	**no ip address** **Example:** `Device(config-if)# no ip address`	Disables IP processing.
Step 16	**service instance** *id* **ethernet** *evc-id* **Example:** `Device(config-if)# service instance 100 ethernet evc_100`	Specifies an Ethernet service instance on an interface and enters service instance configuration mode.

	Command or Action	Purpose
Step 17	**encapsulation dot1q** *vlan-id* **Example:** `Device(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:** `Device(config-if-srv)# bridge-domain 100`	Establishes a bridge domain.
Step 19	**cfm mep domain** *domain-name* **outward mpid** *mpid-value* **Example:** `Device(config-if-srv)# cfm mep domain CUSTOMER outward mpid 1001`	Configures a MEP for a domain.

Step 20	**end**
	Example:
	`Device(config-if-srv)# end`
		Returns to privileged EXEC mode.

Step 21	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 22	**interface** *type name* **Example:**	Specifies an interface and enters interface configuration mode.
Step 23	**no ip address** **Example:** `Device(config-if)# no ip address`	Disables IP processing.
Step 24	**ethernet cfm mip level** *level-id* **Example:** `Device(config-if)# ethernet cfm mip level 7`	Provisions a MIP at a specified maintenance level on an interface.

	Command or Action	Purpose
Step 25	**service instance** *id* **ethernet** *evc-id* **Example:** `Device(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:** `Device(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:** `Device(config-if-srv)# bridge-domain 100`	Establishes a bridge domain.
Step 28	**cfm mep domain** *domain-name* **inward mpid** *mpid-value* **Example:** `Device(config-if-srv)# cfm mep domain PROVIDER inward mpid 201`	Configures a MEP for a domain.

Step 29	**end**
	Example:
	`Device(config-if-srv)# end`
		Returns to privileged EXEC mode.

Step 30	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 31	**ethernet cfm cc enable level** *level-id* **evc** *evc-name* **Example:** `Device(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:** `Device(config)# ethernet cfm cc level any evc evc_100 interval 100 loss-threshold 2`	Sets the parameters for CCMs.

	Command or Action	Purpose

Step 33	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

What to Do Next

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 an 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

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Device(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:** `Device(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:** `Device(config-ether-cfm)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 6	**ethernet evc** *evc-id* **Example:** `Device(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:
	`Device(config-evc)# oam protocol cfm svlan 10 domain cstmr1`
		Configures the EVC OAM protocol.

Step 8	**exit** **Example:** `Device(config-evc)# exit`	Returns 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:
	`Device(config)# end`
		Returns 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:**	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:** `Device(config-if)# ethernet oam max-rate 50`	Enables Ethernet OAM on an interface.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Returns 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
!!
ethernet cfm global
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 global
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!

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 global
!

ethernet cfm mip level 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 global
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!

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 global
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!

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 global
!

ethernet cfm mip level 2
!

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 global
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!

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
!!
ethernet cfm global
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

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
service Customer1 evc evc1 vlan 100

!
```

```
ethernet cfm global
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!

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 MetroCustomer10pA evc evc1 vlan 100
!
ethernet cfm domain OperatorA level 1
mep archive-hold-time 65
 service MetroCustomer10pA evc evc1 vlan 100
!
ethernet cfm global
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!

ethernet cfm mip level 7
ethernet cfm mep level 4 mpid 401 vlan 100
ethernet cfm mep level 1 mpid 101 vlan 100
!

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 MetroCustomer10pA evc evc1 vlan 100
!
ethernet cfm global
!

ethernet cfm mip level 1
!

ethernet cfm mip level 1
```

N-PE A

```
!
ethernet cfm domain ServiceProvider level 4
mep archive-hold-time 60
service MetroCustomer1 evc evc1 vlan 100
!
ethernet cfm domain OperatorA level 1
mep archive-hold-time 65
service MetroCustomer10pA evc evc1 vlan 100
!
ethernet cfm global
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!

ethernet cfm mip level 1
!

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 evc evc1 vlan 100
!
ethernet cfm domain OperatorB level 2
mep archive-hold-time 65
service MetroCustomer1OpB evc evc1 vlan 100
!
ethernet cfm global
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!

ethernet cfm mip level 7
ethernet cfm mep level 4 mpid 402 vlan 100
ethernet cfm mep level 2 mpid 201 vlan 100
!

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 evc evc1 vlan 100
!
ethernet cfm global
!

ethernet cfm mip level 2
!

ethernet cfm mip level 2
N-PE B
!
ethernet cfm domain ServiceProvider level 4
mep archive-hold-time 60
service MetroCustomer1 evc evc1 vlan 100
!
ethernet cfm domain OperatorB level 2
mep archive-hold-time 65
service MetroCustomer1OpB evc evc1 vlan 100
!
ethernet cfm global
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!

ethernet cfm mip level 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
service Customer1 vlan 100
!
ethernet cfm global
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!

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
```

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.

CHAPTER 7

Ethernet Performance Monitoring on Untagged EFPs

The Ethernet Performance Monitoring on untagged EFPs feature enables sessions to run on untagged Ethernet flow points (EFPs).

Information about Ethernet Performance Monitoring on Untagged EFPs

Untagged EFPs

The Ethernet Performance Monitoring on untagged EFPs feature enables sessions to run on untagged Ethernet flow points (EFPs). If an EFP is configured as untagged, then the EFP handles any frames without a dot1q tag, that it receives. Any frames sent using this EFP do not have a dot1q tag.

The dot1q tag contains class of service (CoS) bits, which are used by EPM to test delay or loss of packets with a specific CoS. This support is unavailable when using EPM over untagged EFPs but all other performance monitoring functionality is supported.

How to Configure Ethernet Performance Monitoring on Untagged EFPs

Configuring Ethernet Performance Monitoring on Untagged EFPs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type/number*
4. **service instance** *ID* **ethernet***evc-id*
5. **encapsulation untagged**
6. **end**
7. **configure terminal**
8. **ip sla** *operation-number*
9. **ethernet y1731** {**delay** | **loss**} *type* **domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | *mac-address target-address*} **cos** *cos-value* {**source** {**mpid** *source-mp-id* | *mac-address tsource-address*}}
10. **exit**
11. **ip sla schedule** *operation-number* **start-time** *time* **life** *life*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type/number* **Example:** `Device(config)# interface GigabitEthernet0/0`	Configures an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**service instance** *ID* **ethernet***evc-id* **Example:** `Device(config-if)# service instance 1 ethernet 50`	Configures a service instance and enters service instance configuration mode.
Step 5	**encapsulation untagged** **Example:** `Device(config-if-srv)# encapsulation untagged`	Sets the encapsulation as untagged.

Step 6	**end**
	Example:
	`Device(config-if-srv)# end`
		Returns to privileged EXEC mode.

Step 7	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 8	**ip sla** *operation-number* **Example:** `Device(config)# ip sla 501`	Configures a Cisco IOS IP Service Level Agreements (SLAs) operation and enter IP SLA configuration mode.

Step 9	**ethernet y1731** {**delay** | **loss**} *type* **domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | *mac-address target-address*} **cos** *cos-value* {**source** {**mpid** *source-mp-id* | *mac-address tsource-address*}}
	Example:
	`Device (config-ip-sla)# ethernet y1731 delay DMM domain domain1 evc evc1 mpid 101 cos 0 source mpid 100`
		Begins configuring the receiver on the responder and enters IP SLA Y.1731 delay configuration mode.
		• The source-mp-id or source-address configured by this command corresponds to that of the MEP being configured.
		Note The type argument in the above command syntax takes the following values: DMM, SLM.

Step 10	**exit** **Example:** `Device (config-ip-sla)# exit`	Exits IP SLA configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 11	**ip sla schedule** *operation-number* **start-time** *time* **life** *life* **Example:** `Device(config-sla-y1731-delay)# ip sla schedule 501 start-time now life forever`	Begins a probe with a specified operation number starting at the specified timestamp (or ‘now’ for immediately) for the specified lifetime in seconds (or ‘forever’ to run until the configuration is removed).

Step 12	**end**
	Example:
	`Device(config-sla-y1731-delay)# end`
		Returns to privileged EXEC mode.

Verifying Ethernet Performance Monitoring on Untagged EFPs

Perform the following task to verify the Ethernet Performance Monitoring on Untagged EFPs

SUMMARY STEPS

1. Enter the **show ip sla statistics** to display performance monitoring sessions with untagged EFPs.

DETAILED STEPS

Enter the **show ip sla statistics** to display performance monitoring sessions with untagged EFPs.

Example:

```
Device# show ip sla statistics
IPSLAs Latest Operation Statistics

IPSLA operation id: 5
Loss Statistics for Y1731 Operation 5
Type of operation: Y1731 Loss Measurement
Latest operation start time: *09:08:29.825 PST Wed Jun 11 2014
Latest operation return code: OK
Distribution Statistics:

Interval
 Start time:  *09:08:29.825 PST Wed Jun 11 2014
 Elapsed time: 9 seconds
 Number of measurements initiated: 8
 Number of measurements completed: 8
 Flag: OK
```

Example for Configuring Ethernet Performance Monitoring on Untagged EFPs

Example: Example for Configuring EPM Untagged EFPs

```
Device> enable
Device# configure terminal
Device(config)# interface GigabitEthernet0/0
Device(config-if)# service instance 1 ethernet
Device(config-if-srv)# encapsulation untagged
Device(config-if-srv)# end
Device# configure terminal
Device(config)# ip sla 501
Device(config-ip-sla)# ethernet y1731 delay DMM domain domain1 evc evc1 mpid 101 cos 0
source mpid 100
Device(config-sla-y1731-delay)# exit
Device(config)# ip sla schedule 501 start-time now life forever
Device(config)# end
```

Additional References for Ethernet Performance Monitoring on Untagged EFPs

Related Documents

Related Topic	Document Title
Carrier Ethernet Command Reference	*Cisco IOS Carrier Ethernet Command Reference*
Cisco IOS Master Command List	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*
IP SLAs for Metro Ethernet	"IP SLAs for Metro Ethernet"

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 Ethernet Performance Monitoring on Untagged EFPs

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 10: Feature Information for Ethernet Performance Monitoring on Untagged EFPs

Feature Name	Releases	Feature Information
Ethernet Performance Monitoring on Untagged EFPs	Cisco IOS Release15.5(2)S	The Ethernet Performance Monitoring on untagged EFPs feature enables sessions to run on untagged Ethernet flow points (EFPs). This feature is enabled on Cisco Aggregation Services ASR 903 Series Routers. No commands were introduced or modified.

CHAPTER 8

Syslog Support for Ethernet Connectivity Fault Management

The Cisco 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 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 caveats and feature information, see Bug Search Tool and 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 module.

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 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

- 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

Connectivity Fault 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm logging** [**ais** | **alarm** {**cisco** | **ieee**}] **Example:** `Device(config)# ethernet cfm logging`	Enables all CFM syslog messages.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Disabling 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**no ethernet cfm logging** [**ais** | **alarm** {**cisco** | **ieee**}] **Example:** `Device(config)# no ethernet cfm logging`	Disables all CFM syslog messages.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns 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 connectivity fault management (CFM) syslog messages:

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

Example: Enabling Cisco MIB Syslog Messages

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

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

Example: Enabling IEEE MIB Syslog Messages

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

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

Example: Enabling CFM AIS Syslog Messages

The following example shows how to enable syslog messages specific to the connectivity fault management (CFM) AIS feature:

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

Example: Disabling All CFM Syslog Messages

The following example shows how to disable all connectivity fault management (CFM) syslog messages:

```
Device> enable
Device# configure terminal
Device(config)#no ethernet cfm logging
Device(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 Command 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 11: Feature Information for Syslog Support for Ethernet Connectivity Fault Management

Feature Name	Releases	Feature Information
Syslog Support for Ethernet Connectivity Fault Management	Cisco IOS XE Release 3.8S	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**.

CHAPTER 9

Dynamic Ethernet Service Activation

The Dynamic Ethernet Service Activation (DESA) feature enables the dynamic provisioning of Layer 2 services and transport using dynamic policy. DESA enables increased intelligence in the network control plane, which lowers the cost of network management systems and achieves the following:

- Advanced Ethernet services, with automated subscriber to retail service provider transport mapping and subscriber access service level agreement (SLA) configuration.
- Automated Ethernet services, zero-touch billable Ethernet VPNs and transport services.
- Enhanced retailer network-to-network interface (NNI) that enables enhanced transparency and scalability.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 Dynamic Ethernet Service Activation

- Understanding of how to configure the Ethernet virtual connection (EVC), Accounting, Authentication, and Authorization (AAA), and the Intelligent Services Gateway (ISG) control policies.
- Understanding of how to configure xconnect to configure virtual private wire services (VPWS).
- Cisco 7600 routers with ES+ line cards.

Restrictions for Dynamic Ethernet Service Activation

- Static pseudowires cannot be configured from RADIUS.
- A physical interface can support a maximum of 100 Layer 2 contexts.
- A dynamic service instance identifier must begin at 101.
- Security access control lists (ACLs) are not supported. ACL definitions are defined in the user profile.
- Manual or static configuration cannot be applied to a dynamic Ethernet session after the configuration is downloaded.
- The connectivity fault management (CFM) domain cannot be downloaded from an AAA server. CFM domains must be configured on the router prior to EVC download.
- Dynamic creation of a switched virtual interface (SVI) from AAA is not supported
- Dynamic creation of virtual private LAN service (VPLS) virtual forwarding instance (VFI) and SW-based Ethernet over MPLS (EoMPLS) (that is, xconnect under SVI) from AAA is not supported.
- CISCO-EVC-MIB is not supported for the dynamic Ethernet sessions.
- Per-flow (traffic class) Ethernet accounting is not supported.
- VPLS cannot be configured from RADIUS.

Information About Dynamic Ethernet Service Activation

Overview on Dynamic Ethernet Service Activation

Carrier Ethernet enables service providers to offer ubiquitous end-to-end services and transport mechanisms to their customers.

End-to-end services are categorized as follows:

- Layer 2: L2VPN services
- Layer 3: IP (Internet) or Layer 3 (L3) VPN

Transport mechanisms refer to the technology used by the service provider. Some of the transport mechanism are as follows:

- Native Ethernet
- IP/Multi Protocol Label Switching (MPLS)
- SONET, ATM, Frame Relay (FR), and so on

DESA provides network-based service control by integrating the Cisco EVC framework with a dynamic policy.

DESA delivers an intelligent transport-aware service gateway that can be used at various points in a network. Some of the capabilities provided by DESA are as follows:

- Utilizes AAA to dynamically discover and associate a network transport service with a subscriber context, based on subscriber identity.
- Offers subscriber session awareness at Layer 2.
- Utilizes the ability of the ISG to dynamically apply per-subscriber services based on the subscriber identity, service policy, and subscriber profile derived from the service control layer.
- Provides an abstraction for EVC service configuration above the underlying Ethernet technology, alongside ISG policies and services, with these being subject to be applied or modified based on control and policy plane decisions.

In Cisco IOS Release 15.1(2)S, DESA supports two major functions, EVC accounting and Dynamic Ethernet Layer 2 session provisioning.

EVC Accounting

In a service provider network, billing servers receive accounting records from network elements to measure the usage of particular services by specific users. Billing systems use these records to generate per-usage bills for customers. These accounting records carry traffic statistics measured at a point of interest in the network.

The EVC accounting feature exposes native Ethernet traffic to billing systems via accounting interfaces and policies. Through the integration of EVC, ISG, and AAA functions, Ethernet accounting provides a mechanism for service providers to track usage-based services, billing mechanisms for incremental or temporary services, and provide a traceable accountability method for SLA enforcement.

Ethernet flows between subscriber sites across the Carrier Ethernet network are delivered over an EVC architecture construct. EVC denotes an end-to-end connection across the network on which the user can apply a set of services. Ingress Ethernet frames on a port are mapped or classified to an Ethernet service instance based on the information in the Ethernet frame header. The accounting statistics per Ethernet service instance, which represent aggregate counts for an EVC’s traffic, are collected. The Ethernet service instance represents only one instance of an EVC per port.

Each accounting record includes the following packet information:

- Input packets
- Output packets
- Input bytes
- Output bytes

Ethernet accounting applies to the following connection topologies:

- Point-to-point (P2P)
- Point-to-multipoint
- Multipoint-to-multipoint

Ethernet accounting applies to the following data-forwarding types:

- EVC switched service (EVC over a bridge domain)
- EVC switched service (local switching)
- EVC tunneled service (EVC over MPLS/IP P2P pseudowire (PW))

Ethernet Accounting Configuration

To configure Ethernet accounting, you must first configure accounting traffic classifiers via a class-map policy and associate it with a control policy. Next, you must configure the control policy at the global level, interface level, and dynamic Ethernet session target level. If control policies are configured at multiple levels, the control policy at the inner level has higher precedence over those at higher levels.

The following session-level traffic classification can be applied through the **encapsulation** command:

- Stacked-VLAN (S-VLAN) range or list
- Customer-VLAN (C-VLAN) range or list
- CoS range or list
- VLAN Ethertype
- Payload type

For service instances configured statically via the command-line interface (CLI), you must use the **ethernet subscriber static** command before enabling EVC accounting on the service instance. Without this configuration, the EVC accounting feature cannot be applied.

Per-Session Accounting

Per-session accounting generates a single accounting record for aggregate traffic. This Ethernet ISG session can be either statically or dynamically instantiated.

You can enable accounting at multiple configuration sources such as a user profile on the AAA server, service profile on the AAA server, or service policy on the ISG device. Usage of the ISG control policy for static Ethernet sessions ensures that the steps for enabling per-session accounting remain the same for both static and dynamic Ethernet sessions.

Per-Session RADIUS Accounting Record Format

DESA provides support for generating RADIUS accounting records on a per-subscriber and on a per-class-per-subscriber basis for static and dynamic Ethernet sessions.

Each per-session accounting record can be identified by a unique Acct-Session-ID. The DESA feature introduces two new attributes--stag-vlan-id and ctag-vlan-id. These two new attributes can represent a single or a range of VLAN values.

For detailed steps on configuring Ethernet accounting, see the How to Configure Dynamic Ethernet Service Activation section.

Ethernet Layer 2 Session Provisioning

DESA supports static (preconfigured) and dynamic (dynamic service instances) Ethernet sessions.

Static Ethernet Session Provisioning

Static Ethernet sessions are configured by applying the **ethernet subscriber static** command to Ethernet service instances that are explicitly provisioned using the CLI. DESA supports the application of certain features dynamically to static Ethernet sessions.

Dynamic Ethernet Session Provisioning

Prior to the introduction of DESA, Ethernet service instances had to be configured statically using the CLI. DESA supports creation of dynamic service instances. This dynamic service instance creation is controlled by ISG infrastructure. ISG sessions for Ethernet service instances are referred to as dynamic Ethernet sessions.

DESA provides mechanisms for establishing dynamic Ethernet sessions through an embedded policy plane. The policy plane provides the infrastructure for managing the lifecycle of a session, focusing on authenticating and authorizing sessions.

Dynamic Ethernet sessions are transient in nature, that is, they support start and end events. The start event is marked by the receipt of a frame of interest, which is called the first sign of life (FSoL). The end event is triggered by the expiry of a session idle timer. The FSoL trigger causes a chain of events that starts with subscriber authentication and authorization, followed by service and features determination according to policy rules, thereby leading to dynamic session provisioning and feature or service enablement.

Control Policies

An ISG control policy defines actions that are taken in response to specified events and conditions. Control policies consist of one or more control policy rules. Each control policy rule consists of a condition defined by a control class, session events, and one or more actions. For more information about control policies, refer to the *Cisco IOS Intelligent Services Gateway Configuration Guide* .

You can specify ISG control policies in a hierarchical manner. DESA introduces a new level called the service instance level. For a given session, the policy manager executes the control policy with the highest precedence.

Layer 2 Context

Prior to the introduction of DESA, support for creating service instances was available under Ethernet ports, and you could define only one control policy and one type of session initiator under a single port.

DESA supports the Layer 2 context, a specific Ethernet service instance that classifies FSoL frames and sends them to the device CPU for processing. This processing involves determining whether the FSoL frame should trigger the creation of a dynamic Ethernet session based on AAA authorization.

The Layer 2 context can dynamically trigger multiple service instances based on the configuration within the Layer 2 context. The encapsulation criteria associated with the Layer 2 context must be broad enough to attract desired FSoLs that can trigger dynamic Ethernet sessions.

You can create a new Layer 2 context under an Ethernet port in the following scenarios:

- If there is a requirement to create Ethernet sessions based on multiple different initiators, you can create one Layer 2 context for each type of initiator.
- If there is a requirement to apply different ISG control policies to control sessions under the same Ethernet port.

The number of Layer 2 contexts can be of the same order of magnitude as the number of the ports in the system.

Dynamic Ethernet sessions can be categorized according to the type of the service delimiter that is used to classify (demultiplex) frames into subscriber sessions. In Cisco IOS Release 15.1(2)S, VLAN sessions are the type of dynamic Ethernet sessions supported.

VLAN Sessions

A VLAN session is a dynamic Ethernet session in which the service delimiter is either a VLAN (S-VLAN or C-VLAN), or a VLAN stack; that is, double tagged (S-VLAN + C-VLAN).

Single VLAN

When the service delimiter is a single VLAN, the associated EtherType can be one of the following:

- 0x8100
- 0x9100
- 0x9200
- 0x88a8

You can configure the device with a static Layer 2 context that covers a list or range of single VLANs. There may be multiple Layer 2 contexts per interface (with disjoint VLAN sets). VLAN sessions are logically instantiated over the context with the matching encapsulation. There can be multiple sessions over a single Layer 2 context.

VLAN Stack

When the service delimiter is a VLAN stack, the outermost VLAN can have any of the EtherTypes presented under the single VLAN section, whereas the inner VLAN must have an EtherType of 0x8100. The device can be configured with a static Layer 2 context that matches a unique outermost tag (S-VLAN) and a range of inner tags (C-VLANs).

There can be many static Layer 2 contexts per physical port with nonoverlapping encapsulation. VLAN sessions are logically instantiated over the context with appropriate encapsulation. There can multiple sessions over one Layer 2 context.

There is always a unique session per C-VLAN within a given S-VLAN.

FSoL Detection for AToM VC

DESA enables dynamic EoMPLS virtual circuits (VCs) to be established upon the receipt of FSoL events on the pre-established LDP session, between the aggregation and distribution nodes.

The FSoLs are gleaned for authorization keys that are sent to the ISG policy plane for downloading the provisioned profiles. This results in the ingress VC being accepted and in the creation of the Ethernet session and the egress VC.

FSoL Mechanisms

DESA enables establishing the dynamic Ethernet sessions upon the receipt of FSoL frames from the access or core side of the Layer 2 Ethernet network.

Various mechanisms can be used by a provider device as the FSoL indication of an incoming Ethernet session from a CE node.

Cisco IOS Release 15.1(2)S supports two types of FSoL mechanisms, unclassified service frames and Label Distribution Protocol (LDP) for Any Transport over MPLS (AToM).

Unclassified Service Frames

Unclassified service frames are frames that do not belong to an existing active session but that trigger dynamic Ethernet sessions. Unclassified frames depend on the following characteristics:

- Type of Ethernet session. Cisco IOS Release 15.1(2)S supports VLAN sessions.
- Classifier of the associated Layer 2 context.

If the Layer 2 context classifier matches a range or list of single VLANs, the FSoL for the Layer 2 session is the first Ethernet frame received on a given VLAN within the range or list.

If the Layer 2 context classifier matches a single S-VLAN and range or list of C-VLANs, the FSoL is receipt of a double-tagged Ethernet frame whose C-VLAN does not have an existing session. That is, the FSoL is the first received frame on the C-VLAN for that S-VLAN.

LDP for AToM

In an MPLS aggregation network, when a service needs to be established based on the dynamic indication coming in from the MPLS core, the MPLS provider edge (PE) device treats AToM LDP VC label advertisements as FSoL. DESA supports an equivalent of Layer 2 context to provide granular control over the LDP FSoL.

The LDP FSoL contexts serve the following two purposes:

- To identify a range of LDP VC label advertisements to initiate or accept dynamic session creation.
- To specify the control policy to be applied for sessions initiated in the context.

Any network element that should accept LDP VC label advertisements as FSoL indications should have already established targeted LDP sessions over which the label advertisements are to be processed. You can set up this targeted LDP session via static configuration.

If a VPWS PE receives the AToM LDP FSoL, a PW is established toward the MPLS aggregation network (using the VC ID and target peer IP address that are gleaned from the FSoL). This PW is associated with a native Ethernet attachment circuit (AC) specified in the RADIUS authorization response. The AC is effectively a dynamic Ethernet session whose attributes are supplied via RADIUS.

When an xconnect is not configured and an LDP VC label advertisement message arrives, based on the host address, the network address, and the VC ID of the peer, an attempt is made to identify a service authorization group. The message is treated as a FSoL only when a match is found for the message, and a request is sent to the policy plane for subscriber authorization. However, if a match is not found, subscriber authorization is not attempted.

When a label withdraw message is received, the system checks for a corresponding xconnect. If the xconnect is found, it is removed. Xconnect is not destroyed in response to a pseudowire status message.

Dynamic Transport Provisioning

Dynamic Ethernet sessions are established when the FSoL events are triggered. The information or metadata provided by the FSoL is used as the authorization keys to download RADIUS profiles for Layer 2 transport or PW session attributes.

Single-Sided Model

DESA supports the single-sided model for Layer 2 VPN (L2VPN) provisioning. In the single-sided model, L2VPN is provisioned on the PE only when deemed necessary. The initiator PE detects and instigates the PWs, while the peer PE authorizes and accepts it. This assumes that a target LDP session has already been established between the two PEs.

Automated Transport Setup VPWS

Upon the creation of dynamic Ethernet sessions on the ingress side, the authorized profile also configures the AToM VPWS as the transport service, which in turn configures the Layer 2 tunnel. All the relevant configuration elements must be present in the authorized profiles.

Dynamic Forwarding Services

Dynamic Ethernet sessions must be associated with a forwarding service in order to complete the service transport setup.

DESA supports the following forwarding services:

- Bridge domain service (native Ethernet multipoint bridging)
- Local connect service (P2P stitched services)
- EoMPLS service (P2P tunneled services)

The forwarding services may or may not be preprovisioned on the device. Either way, the forwarding service is associated with the Ethernet sessions dynamically based on the policy determination. If the forwarding service is not preprovisioned, then it is constructed on the go and bound to the session.

Bridge Domain Service

The bridge domain service is a Layer 2 Ethernet multipoint bridging service. DESA supports the association of a dynamic Ethernet session with a Layer 2 bridge domain. The bridge domain may be preconfigured on the device, or dynamically created based on AAA profiles.

Multiple dynamic Ethernet sessions can share the same bridge domain, or they can each have dedicated bridge domains depending on the AAA profiles.

Local Connect Service

The local connect forwarding service is a P2P service. DESA supports the establishment of a local connect service between two dynamic Ethernet sessions.

Note You cannot set up a local connect service between a dynamic Ethernet session and a static session (or a CLI-configured service instance)

EoMPLS Forwarding Service

The EoMPLS forwarding service enables next-generation wholesale models for service providers, where PWs are used to backhaul services from the subscriber edge to the retailer edge.

EoMPLS does not have the mechanisms for signaling tunnels and sessions within tunnels independently. Instead, the signaling involves the establishment of the PW that traditionally has a one-to-one mapping to a service. That is, each service has a dedicated PW.

However, in EoMPLS, the PW is used to multiplex and backhaul the traffic of several subscribers from the subscriber edge to the retailer edge. It is possible to have a single PW per retailer, or a single PW per access node per retailer.

The provisioning of these PWs can follow one of the following two models:

- Single-sided provisioning
- Double-sided provisioning

In the single-sided provisioning model, the FSoL arrives on the attachment circuit of only one MPLS PE device. LDP is used as the FSoL to trigger the PW setup on the far-end PE over the MPLS core.

In the double-sided provisioning model, the FSoL arrives on the attachment circuits of both MPLS PE devices. LDP is not used as the FSoL to trigger the PW setup over the MPLS core.

Dynamic Ethernet Session Mapping to IP L3VPN

For Ethernet transport of business L3VPN services and for residential 3-play services, support for termination of dynamic Ethernet sessions into IP/L3VPN is required.

Dynamic Ethernet sessions are created on the Ethernet interface and associated with bridge domains. An SVI (interface VLAN) is statically preconfigured with the same identifier as that of the bridge domain, and this SVI is configured with an IP address and optionally a virtual routing and forwarding (VRF) instance. This SVI can then be used to offer Layer 3 termination; for example, business L3VPN services or residential IPTV or video on demand (VoD).

Dynamic Ethernet Session Attributes Features and Control Protocols

After a dynamic Ethernet session is created, you can associate attributes, features, and protocols to the session. This can be done either during the initial session setup phase, or later on in the lifetime of the session via a RADIUS CoA.

DESA Attributes Supported During the Initial Setup Phase and via RADIUS CoA

Quality of Service

The dynamic Ethernet sessions support the dynamic configuration of Modular QoS CLI (MQC) Quality of Service (QoS).

MQC is a CLI structure that allows users to create traffic policies and attach these policies to interfaces. A traffic policy contains a traffic class and one or more QoS features. A traffic class is used to select traffic, while the QoS features in the traffic policy determine how to treat the classified traffic. The QoS policies define the corresponding EVC bandwidth profile and guarantees the negotiated customer SLAs. For more information about MQC QoS, see Applying QoS Features Using the MQC .

Accounting

DESA supports session accounting. Session accounting is used to report information about a session's state.

For more information about session accounting and class-based accounting, see theISG RADIUS Interface chapter of the *Cisco IOS ISG RADIUS CoA Interface Guide* .

Idle Timeout and Session Timeout

The idle timeout feature allows the automatic termination of a dynamic Ethernet session after a period of inactivity. The device monitors the traffic transmission activity of the session, and if a user-specified period of time elapses before any new packets are received or transmitted for a given dynamic Ethernet session, then that session is torn down and its associated resources are freed. This feature allows network operators to protect the device from resource depletion when the sessions are short-lived or transient in nature. The idle timeout period is configured via AAA attributes.

ACLs

Dynamic Ethernet sessions support configuration of Layer 2 and Layer 3 ACLs. Because ACLs can be highly tailored to the services offered, dynamic Ethernet sessions support building the ACL definition dynamically. That is, the global ACL definition can be preconfigured statically on the device or can be downloaded via RADIUS.

DESA Attributes Supported During the Initial Setup Phase

EVC and EVC Per UNI Attributes

The following session attributes are provisioned dynamically upon the initial authorization:

- EVC name
- Encapsulation
- Rewrite (that is, VLAN translations)
- User Network Interface (UNI) count and service type (point-to-point or multipoint)

- CE-VLAN to EVC map
- Layer 2 Control Protocol (L2CP) handling

DHCP Snooping

Dynamic Ethernet sessions support configuration of DHCP snooping on bridge domains. DHCP snooping is a DHCP security feature that provides network security by filtering untrusted DHCP messages. DHCP snooping acts like a firewall between untrusted hosts and DHCP servers. You can use DHCP snooping to differentiate between untrusted interfaces connected to the end user and trusted interfaces connected to the DHCP server or another switch.

The function of DHCP snooping is to watch for DHCP request and response packets. By gleaning data from these packets, a table of MAC interface bindings, also called as DHCP snooping table, is built. These bindings can then be used to validate transactions from other services. For example, IP source guard uses the DHCP snooping bindings to prevent IP address spoofing.

DHCP Snooping Option-82 Data Insertion

In residential, metropolitan Ethernet-access environments, DHCP can centrally manage the IP address assignments for a large number of subscribers. When the DHCP snooping option-82 feature is enabled on the router, a subscriber device is identified by the router port through which it connects to the network (in addition to its MAC address). Multiple hosts on the subscriber LAN can be connected to the same port on the access router and are uniquely identified.

Dynamic Ethernet sessions provide the capability to dynamically configure the DHCP Option 82 subscriber ID on a per Ethernet session basis.

IP Source Guard

IP source guard is a security feature that restricts IP traffic on nonrouted, Layer 2 interfaces by filtering traffic based on the DHCP snooping binding database and on manually configured IP source bindings. You can use IP source guard to prevent traffic attacks caused when a host tries to use the IP address of its neighbor.

The dynamic Ethernet sessions support dynamic configuration of IP source guard. When the IP source guard feature is enabled, it blocks all IP traffic on the session except for DHCP packets, which are captured by DHCP snooping. When a CE receives a valid IP address from the DHCP server, an automatic ACL is installed on the session that permits the traffic from that IP address only. Optionally, this ACL may also permit only traffic from the source MAC address gleaned from the DHCP request. All other traffic ingressed on the session, which does not have the matching source IP address, and optionally source MAC address, is blocked. In addition to DHCP snooping binding, IP source guard also filters IP traffic, based on static IP bindings. This allows the feature to operate on sessions where the clients have statically assigned IP addresses.

Dynamic configuration of IP source guard is supported on per-dynamic Ethernet session basis.

MAC Security

You can use MAC security with dynamically learned and static MAC addresses to restrict a port’s ingress traffic by limiting the MAC addresses that are allowed to send traffic into the port. When you assign secure MAC addresses to a secure port, the port does not forward ingress traffic that has source addresses outside the group of defined addresses. If you limit the number of secure MAC addresses to one and assign a single secure MAC address, the device attached to that port has the full bandwidth of the port.

Dynamic Ethernet sessions support dynamic configuration of the EVC MAC security. MAC security has configuration knobs that apply both per dynamic Ethernet session and per bridge domain, and both can be configured dynamically.

Connectivity Fault Management

Carrier Ethernet networks are operated by multiple independent organizations, with restricted management access to each other's equipment. This imposes a new set of Operations, Administration, and Maintenance (OAM) requirements across Carrier Ethernet networks. Ethernet OAM provides tools for monitoring and troubleshooting end-to-end Ethernet services by providing capabilities for detecting, verifying, and isolating connectivity failures in the network.

Connectivity Fault Management (CFM, IEEE 802.1ag) is a service-level OAM protocol that provides tools for monitoring and troubleshooting end-to-end Ethernet services. Cisco IOS E-OAM implementation relies on CFM for end-to-end status of the Ethernet Service across PE devices in the Carrier Ethernet network and updates the CE device via the Ethernet Local Management Interface (E-LMI). The end-to-end connection can from a PE to PE or from a CE to CE. A service can be identified as an S-VLAN or an EVC service.

Dynamic Ethernet sessions support CFM. Activating CFM involves tasks that are performed once at network provisioning time, such as setting up maintenance domains, in addition to tasks that are completed as part of service provisioning.

For both static and dynamic Ethernet sessions, RADIUS-based dynamic provisioning of per-service CFM attributes is supported. These include the following:

- Creating up and down maintenance endpoints (MEPs) (specifying the domain, maintenance point ID (MPID), CoS, alarm delay, reset interval, and notification options)
- Creating maintenance intermediate points (MIPs) (specifying level) or per MA,MIP autocreate option (the global and per-domain MIP autocreate options are specified as part of network provisioning)
- Defining static remote MEP lists, and enabling/disabling remote MEP check
- Defining short MA names
- Defining CFM sender ID
- Specifying maximum number of MEPs per MA
- Enabling/disabling CCM transmission, and defining continuity check interval and loss threshold
- Enabling Alarm Indication Signal (AIS) and specifying AIS options (period, expiry threshold, alarm suppression, and level)
- Enabling LCK and specifying LCK options (period, expiry threshold and level)
- Specifying CFM encapsulation on dynamic Ethernet sessions with ambiguous classifiers
- OAM Interworking options (CFM to E-LMI, 802.3ah to CFM)

For static Ethernet sessions, the keys used as part of the authorization request to obtain the configuration profile hosting the CFM attributes are as follows:

- EVC ID
- Host name or router ID

- Port ID

For dynamic Ethernet sessions, the keys vary depending on the type of FSoL and will in general be equal to or a subset of the keys required for initial session authorization.

E-LMI

E-LMI is an Ethernet OAM protocol. It provides information that enables autoconfiguration of CE devices and provides the status of EVCs for large Ethernet metropolitan-area networks (MANs) and WANs. Specifically, E-LMI notifies a CE device of the operating state of an EVC and the time when an EVC is added or deleted. E-LMI also communicates the attributes of an EVC and a UNI to a CE device.

E-LMI has significance at the UNI between the metro Ethernet network (MEN) and the CE. The protocol serves two functions:

- Provides fault notification from PE to CE (EVC and remote UNI status).
- Provides message formats to allow the automated configuration of the CE remotely from the PE.

DESA supports the dynamic provisioning of the following E-LMI attributes for the purpose of communicating them to the CE:

- EVC ID
- EVC type (P2P or multipoint)
- CE-VLAN/EVC map

AAA Schema for EVC

The AAA schema for EVC ensures that the off-box configuration and accounting of Ethernet services (via RADIUS) is supported for native Ethernet bridged services.

Dynamic Service Activation and Deactivation Using COA

The DESA feature allows administrators to dynamically apply and remove services on existing Ethernet sessions from an external server using a change of authorization (CoA) extension. A service consisting of individual features must be atomic; if any of the constituent features fail, the entire service is removed, leaving the session in the original state.

The following table provides CoA capabilities that are supported by DESA:

Table 12: CoA Capabilities Supported for PEI

CoA Capabilities	Description
Service Activate	Applies a service (a named collection of EVC and ISG features) on an existing session.
Service Deactivate	Removes a service from an existing session.
Session Query	Queries details related to a session from RADIUS.

CoA Capabilities	Description
Session Query for Service Status	Queries the status of a service.

How to Configure Dynamic Ethernet Service Activation

Configuring AAA for Enabling Accounting

Cisco IOS AAA supports six different types of accounting (network, exec, commands, connection, system, and resource), two accounting record types (stop-only, start-stop), and two accounting methods (TACACS+, RADIUS). You can specify these options by defining an AAA method list by using the **aaa accounting** command. For more information, see the Cisco IOS Security Command Reference and the Configuring Accounting chapter of the *Security Configuration Guide: Securing User Services*.

After defining the AAA method list, you can use it to configure accounting by referring to the named method list from different configuration sources such as the RADIUS user profile, RADIUS service profile, and on-router service policy map.

You can define a default method list (a method list with the name “default”). This default method list is automatically applied to all sessions except those that have a named method list explicitly configured.

Configuring AAA Enabling Interim Accounting Update

You can periodically generate accounting records. Two types of interim accounting are supported, accounting updates for new information and periodic accounting.

Accounting updates for new information can be enabled or disabled globally by issuing the **aaa accounting update** command on a router. However, interval for periodic accounting can be configured at three configuration sources--on the router, in the user profile on the AAA server, and in the service profile on the AAA server. For more information, see the Cisco IOS Security Command Reference and the http://www.cisco.com/en/US/docs/ios/sec_user_services/configuration/guide/sec_cfg_accountg.html“Configuring Accounting” chapter of the *Security Configuration Guide: Securing User Services*.

Configuring ISG Control Policy to Apply ISG Services

To define a control policy, you must first define a control class map to identify events and conditions and then define a control policy map to bind the control class map to different actions. Control polices can be defined in multiple levels such as global, interface, subinterface, virtual-template, VC, and private virtual circuit (PVC).

The policy manager executes the rules in the control policy only after the session comes into existence. For information about configuring control policies, see Configuring ISG Control Policies .

Configuring Per-Session Accounting

Perform the following task to configure per-session accounting for an Ethernet session.

Before You Begin

- Accounting traffic classifiers must be configured via class-map policies and associated with a control policy. For more information about configuring traffic classifiers, see Configuring ISG Control Policies.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance** *id* **ethernet** [*evc-name*]
5. **encapsulation dot1q**
6. **service-policy type control** *policy-map-name*
7. **ethernet subscriber static**
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** *type number* **Example:** `Router(config)# interface ethernet 0/0`	Specifies the interface type and number, and enters interface configuration mode.
Step 4	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Router(config-if)# service instance 1 ethernet test`	Configures an Ethernet service instance on an interface, and to enters service instance configuration mode

	Command or Action	Purpose
Step 5	**encapsulation dot1q** **Example:** `Router(config-if-srv)# encapsulation dot1q`	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance.
Step 6	**service-policy type control** *policy-map-name* **Example:** `Router(config-if-srv)# service-policy type control policy1`	Applies a control policy to a context.
Step 7	**ethernet subscriber static** **Example:** `Router(config-if-srv)# ethernet subscriber static`	Creates static sessions for configuring EVC accounting.

Step 8	**end**
	Example:
	`Router(config-if-srv)# end`
		Exits service instance configuration mode and enters privileged EXEC mode.

Disabling Per-Session Accounting Configuration

Tasks for disabling per-session accounting depends on the methodology that was used to configure the per-session accounting feature.

Disabling a per-flow accounting configuration when the feature was installed through a per-user profile

SUMMARY STEPS

1. Modify the user profile associated with the target session to remove the Cisco Attribute Value (AV) pair "accounting-list=method_list_name"
2. Clear the target session by using the CLI command or packet of disconnect (PoD) from the AAA server.

DETAILED STEPS

Step 1 Modify the user profile associated with the target session to remove the Cisco Attribute Value (AV) pair "accounting-list=method_list_name"

Step 2 Clear the target session by using the CLI command or packet of disconnect (PoD) from the AAA server.

Disabling a per-flow accounting configuration when the feature was installed through a service profile

Perform this task to disable a per-flow accounting configuration, if the feature was installed through a service profile on an AAA server or a service-policy on the router.

SUMMARY STEPS

1. Identify the Acct-Session-ID (the RADIUS attribute 44) associated with the target session.
2. Identify the service name (that is, the service-profile name on AAA server or the service-policy name on the router) that contains the feature that must be uninstalled.
3. Apply the Acct-Session-ID and service name with the ISG service deactivate mechanism to remove the service from the target session. This mechanism makes use of the RADIUS CoA feature. For more information, see the *Cisco ISG RADIUS Interface Guide* .

DETAILED STEPS

Step 1 Identify the Acct-Session-ID (the RADIUS attribute 44) associated with the target session.

Step 2 Identify the service name (that is, the service-profile name on AAA server or the service-policy name on the router) that contains the feature that must be uninstalled.

Step 3 Apply the Acct-Session-ID and service name with the ISG service deactivate mechanism to remove the service from the target session. This mechanism makes use of the RADIUS CoA feature. For more information, see the *Cisco ISG RADIUS Interface Guide* .

Note Deactivating a service on a session removes all the features applied through the service.

Modifying Per-Session Accounting Configuration

In an ISG framework, you can activate a feature by configuring it inside a user profile or bundling it inside an off-box service-profile or on-box service-policy.

You can modify a per-session accounting configuration by first deactivating a service, and then reactivating a new service. Alternatively, you can modify the service definition and clear all the sessions using the service to force them to reauthorize.

Configuring a Layer 2 Context

Perform this task to configure a Layer 2 context.

Note

- Only one initiator is allowed in each Layer 2 context.
- Modification of the **encapsulation** command associated with a Layer 2 context causes all the associated dynamic Ethernet sessions to be disconnected.
- You cannot specify an IP subscriber initiator in Layer 2 contexts.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance** *id* **ethernet** [*evc-name*]
5. **encapsulation** {{**dot1ad**| **dot1q**}[*vlan-id*| **any**] {**cos** *cos-value* | **etype** *type*| **exact** | **second-dot1q**| **vlan-type**} | **priority-tagged** [**cos** *cos-value*] [**etype** *type*] | **untagged**[**etype** *type*]}
6. **ethernet subscriber**
7. **initiator unclassified vlan**
8. **service-policy type control** *policy-map-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 0/0`	Specifies the interface type and number, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**service instance** *id* **ethernet** [*evc-name*] **Example:** `Router(config-if)# service instance 1 ethernet test`	Configures an Ethernet service instance on an interface, and to enters service instance configuration mode

Step 5	**encapsulation** {{**dot1ad**| **dot1q**}[*vlan-id*| **any**] {**cos** *cos-value* | **etype** *type*| **exact** | **second-dot1q**| **vlan-type**} | **priority-tagged** [**cos** *cos-value*] [**etype** *type*] | **untagged**[**etype** *type*]}
	Example:
	`Router(config-if-srv)# encapsulation dot1q`
		Specifies the classifier to identify the subset of traffic associated with a port.

Step 6	**ethernet subscriber** **Example:** `Router(config-if-srv)# ethernet subscriber`	Enables the Ethernet Layer 2 context. • This context is used for creating dynamic Ethernet sessions. • To disconnect the existing sessions from the Layer 2 context, use the **no ethernet subscriber** command.
Step 7	**initiator unclassified vlan** **Example:** `Router(config-if-srv)# initiator unclassified vlan`	Enables an initiator for detecting the FSoL under Ethernet Layer 2 context. • You must remove the existing initiators before configuring new ones. • To remove an existing initiator, use the **no initiator unclassified vlan** command.
Step 8	**service-policy type control** *policy-map-name* **Example:** `Router(config-if-srv)# service-policy type control policy1`	(Optional) Applies a control policy to a context. • If a control policy is not specified, then the policy manager attempts to find one defined in the hierarchy. • To remove an existing control policy, use the **no service-policy type control** command.

Step 9	**end**
	Example:
	`Router(config-if-srv)# end`
		Exits service instance configuration mode and enters privileged EXEC mode.

Enabling Dynamic Ethernet Sessions

Perform the following task to enable dynamic Ethernet sessions:

SUMMARY STEPS

1. Configure an Layer 2 context on the router. For more information about configuring an Layer 2 context, see the Configuring a Layer 2 Context, on page 296.
2. Configure session keys using a control policy on the router. For more information on configuring session keys by using control policies, see Configuring ISG Control Policies .
3. Configure a per-user profile for dynamic Ethernet sessions on the AAA server. Every dynamically instantiated Ethernet session must have a unique per-user profile on the AAA server. A per-user profile on the AAA server is essentially a set of AAA attributes identified by a username. In case of DESA, the username for the per-user profile is constructed from the session keys.
4. Configure a service profile for the required forwarding services. Cisco recommends that you define a different profile for the forwarding service and use the forwarding service AAA attribute to tie both the profiles.
5. Configure a service profile for the desired EVC and ISG services.

DETAILED STEPS

Step 1 Configure an Layer 2 context on the router. For more information about configuring an Layer 2 context, see the Configuring a Layer 2 Context, on page 296.

Step 2 Configure session keys using a control policy on the router. For more information on configuring session keys by using control policies, see Configuring ISG Control Policies .

Step 3 Configure a per-user profile for dynamic Ethernet sessions on the AAA server. Every dynamically instantiated Ethernet session must have a unique per-user profile on the AAA server. A per-user profile on the AAA server is essentially a set of AAA attributes identified by a username. In case of DESA, the username for the per-user profile is constructed from the session keys.

Step 4 Configure a service profile for the required forwarding services. Cisco recommends that you define a different profile for the forwarding service and use the forwarding service AAA attribute to tie both the profiles.

Step 5 Configure a service profile for the desired EVC and ISG services.

Configuring AToM Subscribers

Perform the following task to configure AToM subscribers:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **l2 subscriber authorization group** *group-name*
4. **peer** {**host** *destination-host-address*| **network** *destination-network-address destination-network-mask*} *vc-id*[*vc-id-range*]
5. **service-policy type control** *policy-map-name*
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	**l2 subscriber authorization group** *group-name* **Example:** `Router(config)# l2 subscriber authorization group group1`	Creates a Layer 2 subscriber authorization group, and enters Layer 2 subscriber group mode. • You must define mutually exclusive service authorization groups.
Step 4	**peer** {**host** *destination-host-address*| **network** *destination-network-address destination-network-mask*} *vc-id*[*vc-id-range*] **Example:** `Router(config-l2-sub-gr)# peer host 10.10.1.1 23 54`	Defines the target LDP peer PE information. • Within a router, the *destination-host-address* and *vc-id-range* combination must be unique to identify a unique service authorization group.
Step 5	**service-policy type control** *policy-map-name* **Example:** `Router(config-if-srv)# service-policy type control policy1`	(Optional) Applies a control policy to a context.

	Command or Action	Purpose

Step 6	**end**
	Example:
	`Router(config-if-srv)# end`
		Exits service instance configuration mode and enters privileged EXEC mode.

Verifying Per-Session Accounting and Layer 2 Context

Perform the following task to verify the per-session accounting configuration:

SUMMARY STEPS

1. Enter the **show subscriber session** command to display information about subscriber sessions on the ISG.
2. Enter the **show aaa sessions** command to display AAA subscriber information, including the unique ID.
3. Enter the **show aaa user** command to display attributes related to the AAA session.
4. Enter the **show ethernet service instance detail** command to display information about Ethernet service instances.
5. Enter the **show mpls l2transport vc detail** command to display information about AToM VCs and static PWs that have been enabled to route Layer 2 packets on a router.
6. Enter the **show xconnect all detail** command to display information about xconnect ACs and pseudowires.

DETAILED STEPS

Step 1 Enter the **show subscriber session** command to display information about subscriber sessions on the ISG.

Example:

```
Router# show subscriber session uid 100 detailed
Subscriber session handle: AAAAAAAA, state: connected, service: xxxx
Unique Session ID: 100
... ... ...
Session inbound features:
Feature: Session accounting
Method List: my_aaa_method_list
Outbound direction:
Packets = 1000 Bytes = 40000
Session outbound features:
Feature: Session accounting
Method List: my_aaa_method_list
Outbound direction:
Packets = 1000 Bytes = 4000
... ... ...
```

Step 2 Enter the **show aaa sessions** command to display AAA subscriber information, including the unique ID.

Example:

```
Router# show aaa user all
Unique id 100 is currently in use.
Accounting:
log=xxxx
Events recorded :
... ... ...
Cumulative Byte/Packet Counts :
Bytes In = 40000 Bytes Out = 40000
Paks In = 1000 Paks Out = 1000
... ... ...
StartTime = xxx
AuthenTime = xxx
    Component = IEDGE_ACCOUNTING
```

Step 3 Enter the **show aaa user** command to display attributes related to the AAA session.

Example:

```
Router# show aaa user all
Unique id 100 is currently in use.
Accounting:
log=xxxx
Events recorded :
... ... ...
Cumulative Byte/Packet Counts :
Bytes In = 40000 Bytes Out = 40000
Paks In = 1000 Paks Out = 1000
... ... ...
StartTime = xxx
AuthenTime = xxx
    Component = IEDGE_ACCOUNTING
```

Step 4 Enter the **show ethernet service instance detail** command to display information about Ethernet service instances.

Example:

```
Router# show ethernet service instance detail
Service Instance ID: 1
Service instance type: L2Context
Intiators: unclassified vlan
Control policy: ABC
Associated Interface: Ethernet0/0
Associated EVC:
L2protocol drop
CE-Vlans:
Encapsulation: dot1q 200-300 vlan protocol type 0x8100
Interface Dot1q Tunnel Ethertype: 0x8100
State: Up
EFP Statistics:
   Pkts In   Bytes In   Pkts Out  Bytes Out
         0          0          0          0
```

Step 5 Enter the **show mpls l2transport vc detail** command to display information about AToM VCs and static PWs that have been enabled to route Layer 2 packets on a router.

Example:

```
Router# show mpls l2transport vc detail
Local interface: Et0/0 up, line protocol up, Eth VLAN 22 up
  Destination address: 33.33.33.34, VC ID: 12346, VC status: up
    Output interface: Et4/0, imposed label stack {19 20}
    Preferred path: not configured
    Default path: active
    Next hop: 11.11.11.12
  Create time: 00:02:23, last status change time: 00:02:23
  Signaling protocol: LDP, peer 33.33.33.34:0 up
    Targeted Hello: 33.33.33.33(LDP Id) -> 33.33.33.34, LDP is UP
    Status TLV support (local/remote)   : enabled/supported
      LDP route watch                   : enabled
      Label/status state machine        : established, LruRru
      Last local dataplane   status rcvd: No fault
      Last BFD dataplane     status rcvd: Not sent
      Last local SSS circuit status rcvd: No fault
      Last local SSS circuit status sent: No fault
      Last local  LDP TLV    status sent: No fault
      Last remote LDP TLV    status rcvd: No fault
      Last remote LDP ADJ    status rcvd: No fault
    MPLS VC labels: local 22, remote 20
    PWID: 8199
    Group ID: local 0, remote 0
    MTU: local 1500, remote 1500
    Remote interface description:
  Sequencing: receive disabled, send disabled
  Control Word: On (configured: autosense)
  VC statistics:
    transit packet totals: receive 0, send 0
    transit byte totals:   receive 0, send 0
    transit packet drops:  receive 0, seq error 0, send 0
```

Step 6 Enter the **show xconnect all detail** command to display information about xconnect ACs and pseudowires.

Example:

```
Router# show xconnect all detail
Legend:    XC ST=Xconnect State  S1=Segment1 State  S2=Segment2 State
  UP=Up       DN=Down            AD=Admin Down      IA=Inactive
  SB=Standby  HS=Hot Standby     RV=Recovering      NH=No Hardware
XC ST  Segment 1                         S1 Segment 2                         S2
------+---------------------------------+--+---------------------------------+--
UP     ac   Et0/0:22(Eth VLAN)           UP mpls 33.33.33.34:12346            UP
            Interworking: ethernet               Local  VC label 22
                                                 Remote VC label 20
                                                 pw-class:
```

Configuration Examples for Dynamic Ethernet Service Activation

Example Configuring AAA Accounting

The following example shows how to configure the resource failure stop accounting and resource accounting for start-stop records functions:

```
!Enable AAA on your network access server.
aaa new-model
!Enable authentication at login and list the AOL string name to use for login
authentication.
aaa authentication login AOL group radius local
!Enable authentication for ppp and list the default method to use for PPP authentication.
aaa authentication ppp default group radius local
!Enable authorization for all exec sessions and list the AOL string name to use for
```

```
authorization.
aaa authorization exec AOL group radius if-authenticated
!Enable authorization for all network-related service requests and list the default method

to use for all network-related authorizations.
aaa authorization network default group radius if-authenticated
!Enable accounting for all exec sessions and list the default method to use for all
start-stop accounting services.
aaa accounting exec default start-stop group radius
!Enable accounting for all network-related service requests and list the default method to

use for all start-stop accounting services.
aaa accounting network default start-stop group radius
!Enable failure stop accounting.
aaa accounting resource default stop-failure group radius
!Enable resource accounting for start-stop records.
aaa accounting resource default start-stop group radius
```

Examples Configuring ISG Control Policy to Apply ISG Services

The following example shows how to enable an ISG control policy that directly applies to a service policy:

```
policy-map type control SampleControlPolicyMap1
    class type control always event session-start
      1 service-policy type service SampleAccountingPolicy
```

The following example shows how to enable an ISG control policy that gets authorizations from the AAA server:

```
policy-map type control SampleControlPolicyMap2
    class type control always event session-start
      1 authorize identifier stag-vlan-id plus cos-vlan-id
```

Example Configuring Per-Session Accounting

The following example shows how to define a control policy on the router:

```
policy-map type control SampleControlPolicyMap
    class type control always event session-start
      1 service-policy type service SampleAccountingPolicy
```

The following example shows how to define a service policy on the router:

```
class-map type traffic match-any EmptyClassMap
policy-map type service SampleServicePolicyMap
   class type traffic EmptyClassMap
     accounting aaa list my-method-list
```

The following example shows how to create a static Ethernet session and associate it with the previously defined control policy:

```
service instance 10 ethernet
   encapsulation dot1q 100
   service-policy type control SampleServicePolicyMap.
   ethernet subscriber static
```

Examples Configuring Service Instances

The following example shows how to configure a static bridge domain. This is an example of native Ethernet, where there is no requirement of creating ISG sessions:

```
interface ethernet 0/0
  service instance dot1q 1 second-dot1q 1-2000
   bridge-domain 100
```

The following example shows how to configure a static Ethernet session. In this case, one ISG session is created for every service instance. The initiator of the ISG session is statically configured.

```
interface ethernet 0/0
  service instance 1 ethernet
     encapsulation dot1q 1 second-dot1q 1-2000
     ethernet subscribers static
     bridge-domain 100
```

The following example shows how to configure a service instance that is treated as a Layer 2 context:

Note Layer 2 context and static Ethernet sessions are mutually exclusive on the same port.

```
interface ethernet 0/0
  service instance 1 ethernet
   encapsulation dot1q 1 second-dot1q 1-2000
   service-policy type control mypolicy
   ethernet subscribers
     initiator unclassified-vlan
```

Example Configuring Layer 2 Context

The following example shows how to create a Layer 2 context of unclassified VLAN type:

```
!!Layer2 Context 2
interface Ethernet 0/0
  service instance 2 ethernet
   encapsulation dot1q 1 second-dot1q 2001-4094
   ethernet subscriber
     initiator unclassified-vlan
```

Example Configuring AToM Subscribers

The following example shows how to configure an AToM subscriber:

```
l2subscriber authorization group list1
  peer host 10.10.1.1 vc-id 100-200
   service-policy type control ldpFSOL-ctrl-policy-1
l2subscriber authorization group list2
  peer network 10.10.2.1 mask 255.255.255.0 vc-id 100-200
   service-policy type control ldpFSOL-ctrl-policy-2
```

Example Configuring Single-Sided Dynamic L2VPN VPWS

The following example shows how to configure dynamic L2VPN VPWS on the PE router:

```
l2 subscriber authorization group atom_test1
 service-policy type control atom_rule1
 peer network 10.10.1.1 255.255.0.0 1 4294967295
```

The following is sample RADIUS peer profile configuration:

```
RADIUS Profile
Peer IP Profile (Username: peer-ip:102.102.102.102:vc-id:111111)
Cisco-AVPair = l2vpn:vcid=111111
Cisco-AVPair = l2vpn:service-id=vpws_pw_customer1
Cisco-AVPair = subscriber:sss-service=vpws
Cisco-AVPair = l2vpn:redundancy-group=2
Cisco-AVPair = l2vpn:pw-encapsulation=mpls
Cisco-AVPair = l2vpn:peer-ip-address=102.102.102.102
```

The following is sample L2VPN profile configuration:

```
(Username: vpws_pw_customer1)
Cisco-AVPair = l2vpn:member=ethernet-service-instance:Gi2/3 -stag-type:0x8100
-stag-vlan-id:1000
Cisco-AVPair = l2vpn:member=pseudowire:peer-ip:102.102.102.102:vc-id:111111
```

The following is sample RADIUS user profile configuration:

```
RADIUS Profile
User Profile (Username: RouterA:nas-port:2/0/3/0:1000)
Cisco-AVPair = subscriber:sss-service=vpws
Cisco-AVPair = l2vpn:redundancy-group=1
Cisco-AVPair = l2vpn:service-id=vpws_pw_customer1
Cisco-AVPair = ethernet-service-instance:service-instance-description=Dynamic customer 1
Cisco-AVPair = ethernet-service-instance:stag-vlan-id=1000
Cisco-AVPair = ethernet-service-instance:rewrite-ingress=1
Cisco-AVPair = ethernet-service-instance:rewrite-ingress-tag-operation=Pop1
Cisco-AVPair = ethernet-service-instance:rewrite-ingress-symmetric=TRUE
```

You can verify the Layer 2 context configuration on the PE router with the **show interface**command, as follows:

```
Router# show interface gigabit ethernet 2/3
interface GigabitEthernet2/3
 service instance dynamic 90 ethernet
  description L2 context for single-tag FSOL
  encapsulation dot1q 1000-2000
  ethernet subscriber
  initiator unclassified vlan
  service-policy type control DYNAMIC_EVC
```

You can verify the dynamic service instance on the PE router with the **show derived-config** command, as follows:

```
Router# show derived-config interface gigabit ethernet 2/3
interface GigabitEthernet2/3
<Output snipped for clarity>
.
.
 service instance 101 ethernet
  description Dynamic customer 1
  encapsulation dot1q 1000
  rewrite ingress tag pop 1 symmetric
  xconnect 102.102.102.102 111111 encapsulation mpls
```

Example Configuring Double-Sided Dynamic L2VPN VPWS

In the double-sided provisioning model, you must configure both MPLS PE devices.

The following example shows how to configure L2VPN VPWS on the PE1 router:

```
l2 subscriber authorization group atom_test1
 service-policy type control atom_rule1
 peer network 10.10.10.2 255.255.0.0 1 4294967295
```

The following example shows how to configure L2VPN VPWS on the PE2 router:

```
l2 subscriber authorization group atom_test1
 service-policy type control atom_rule1
 peer network 10.10.10.1 255.255.0.0 1 4294967295
```

The following example shows how to verify the Layer 2 context configuration on the PE1 router:

```
Router# show interface gigabit ethernet 2/3
interface GigabitEthernet2/3
 service instance dynamic 90 ethernet
  description L2 context for single-tag FSOL
  encapsulation dot1q 1000-2000
  ethernet subscriber
  initiator unclassified vlan
  service-policy type control DYNAMIC_EVC
```

The following example shows how to verify the Layer 2 context configuration on the PE2 router:

```
Router# show interface gigabit ethernet 2/4
interface GigabitEthernet2/4
 service instance dynamic 90 ethernet
  description L2 context for single-tag FSOL
  encapsulation dot1q 1000-2000
  ethernet subscriber
  initiator unclassified vlan
  service-policy type control DYNAMIC_EVC
```

The following example shows how to verify the dynamic service instance on the PE1 router:

```
Router# show derived-config interface gigabit ethernet 2/3
interface GigabitEthernet2/3
<Output snipped for clarity>
.
.
 service instance 101 ethernet
  description Dynamic customer 1
  encapsulation dot1q 1000
  rewrite ingress tag pop 1 symmetric
  xconnect 10.10.10.2 111111 encapsulation mpls
```

The following example shows how to verify the dynamic service instance on the PE2 router:

```
Router# show derived-config interface gigabit ethernet 2/4
interface GigabitEthernet2/4
<Output snipped for clarity>
.
.
 service instance 102 ethernet
  description Dynamic customer 1
  encapsulation dot1q 1000
  rewrite ingress tag pop 1 symmetric
  xconnect 10.10.10.1 111111 encapsulation mpls
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands: master list of commands with complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS Master Commands List, All Releases
Carrier Ethernet commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS Carrier Ethernet Command Reference
ISG commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS Intelligent Services Gateway Command Reference
AAA commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS Security Command Reference
Configuring ISG control policies	Configuring ISG Control Policies module
Configuring different types of AAA accounting	Configuring Accounting module

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 Dynamic Ethernet Service Activation

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 Dynamic Ethernet Service Activation

Feature Name	Releases	Feature Information
Dynamic Ethernet Service Activation	15.1(2)S	The DESA feature enables the dynamic provisioning of Layer 2 services and transport using the dynamic policy. The following commands were introduced or modified: **ais**, **authorize identifier**, **continuity-check**, **debug ethernet service**, **debug ethernet service instance dynamic**, **debug idmgr**, **debug mpls l2transport vc subscriber**, **ethernet cfm mip**, **ethernet subscriber**, **ethernet subscriber session**, **ethernet subscriber static**, **initiator unclassified vlan**, **l2 subscriber**, **maximum meps**, **mep mpid**, **mip auto-create**(cfm-srv), **peer**, **pseudowire** (Layer 2**)**, **service evc**, **service_instance_dynamic**, **service-policy type control policy**, **show database data**, **show derived-config**, **show dwnld_mgr**, **show ethernet cfm domain**, **show ethernet cfm maintenance-points local**, **show ethernet service instance**, **show ethernet service dynamic**, **show subscriber session**.

CHAPTER 10

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 caveats and feature information, see Bug Search Tool and 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 module.

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

EVCs

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 contains 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 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 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 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mac access-list extended** *name* **Example:** `Device(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:
	`Device(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 You Begin

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(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:** `Device(config-if)# service instance 100 ethernet`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.

	Command or Action	Purpose
Step 5	**encapsulation dot1q vlan-id** **Example:** `Device(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	**mac access-group** *access-list-name* in **Example:** `Device(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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mac access-list extended** *name* **Example:** `Device(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:
	`Device(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:
	`Device(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:
	`Device(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:** `Device(config-ext-macl)# deny any any`	Prevents forwarding of Layer 2 traffic except for the allowed ACEs.
Step 8	**exit** **Example:** `Device(config-ext-macl)# exit`	Exits the current command mode and returns to global configuration mode.
Step 9	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet 1/0/0`	Specifies the interface.

	Command or Action	Purpose
Step 10	**service instance** *id* **ethernet** **Example:** `Device(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:** `Device(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:** `Device(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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal **Example:** `Device# 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:** `Device# 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.

```
Device# 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 14: 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.

Field	Description
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 Command List, All Releases
Configuring CFM over an EFP Interface with the Cross Connect feature on the Cisco ASR 903 Router.	Configuring the CFM over EFP Interface with Cross Connect Feature
Configuring Ethernet Virtual Connections on the Cisco ASR 903 Router	Configuring Ethernet Virtual Connections on the Cisco ASR 903 Router

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)*

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 15: Feature Information for Layer 2 Access Control Lists on EVCs

Feature Name	Releases	Feature Information
Layer 2 Access Control Lists on EVCs	Cisco IOS XE Release 3.6S	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**.

CHAPTER 11

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 caveats and feature information, see Bug Search Tool and 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 module.

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

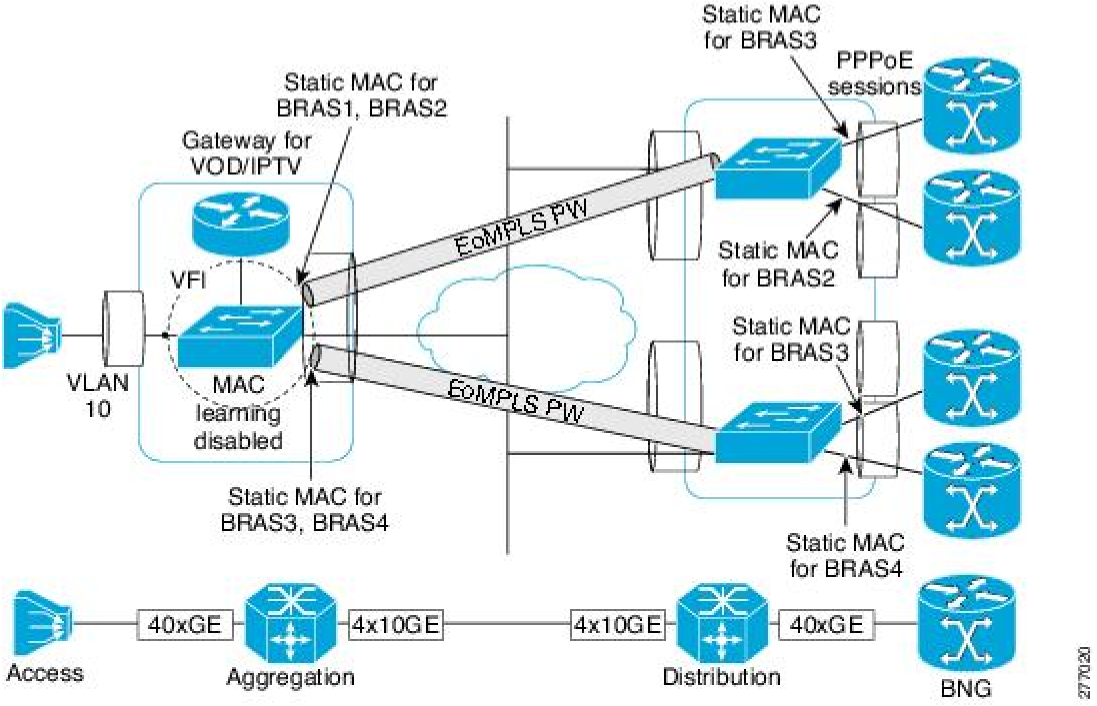
Static MAC Address Support on Service Instances and Pseudowires

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.
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.

	Command or Action	Purpose
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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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 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 16: 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**.

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

CHAPTER 12

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 caveats and feature information, see Bug Search Tool and 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 module.

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *subslot* / *port* [**.***subinterface-number*] **Example:** `Device(config)# interface gigabitethernet 4/0/0`	Specifies the interface to configure and enters interface configuration mode.
Step 4	**service instance** *id* **ethernet** [*evc-id*] **Example:** `Device(config-if)# service instance 101 ethernet`	Creates a service instance (an instance of an Ethernet virtual circuit [EVC]) on an interface and enters service instance configuration mode.
Step 5	**encapsulation dot1q** *vlan-id* [**native**] **Example:** `Device(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.

	Command or Action	Purpose
Step 6	**bridge-domain** *bridge-id* [**split-horizon** [**group** *group-id*]] **Example:** `Device(config-if-srv)# bridge-domain 12`	Binds the service instance to a bridge domain instance.

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:

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

Issue the following command to verify the configuration:

```
Device# 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, Gigabit Ethernet interface 4/0/0 and Gigabit Ethernet interface 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 Multiservice Transport Platform (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:

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

Use the **show spanning-tree vlan** command to verify the configuration, as shown in the following two examples.

```
Device# 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

Device# 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, Gigabit Ethernet interface 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:

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

Now two service instances are configured on Gigabit Ethernet interface4/0/3 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 Gigabit Ethernet interface gi4/0/3 in the MST instance to which it belongs.

Use the**show spanning-tree mst** command to display the information about the Multiple Spanning Tree (MST) protocol, as shown below.

```
Device# 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 Command 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)*

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 17: 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**.

CHAPTER 13

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 caveats and feature information, see Bug Search Tool and 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 module.

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

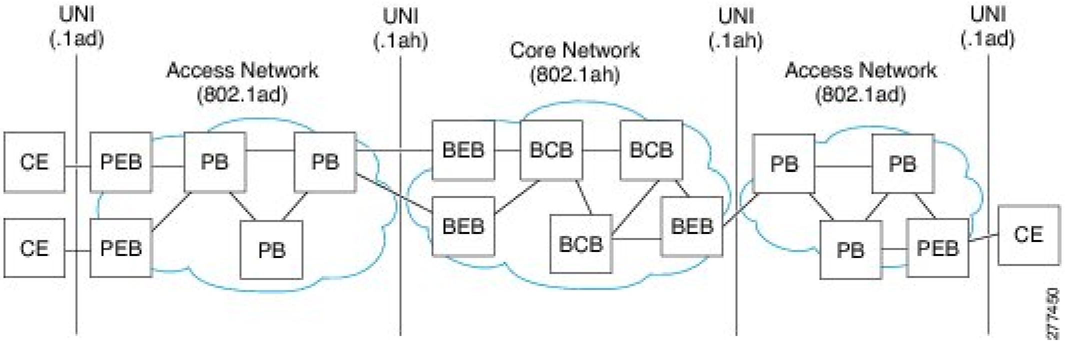
The IEEE 802.1ah on Provider Backbone Bridges feature encapsulates the end users traffic inside the service providers MAC header, enabling the backbone edge bridge (BEB) to support large numbers of service instances. This functionality is also known as MAC-in-MAC or MAC Tunneling Protocol (MTP). It also 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 18: IEEE 802.1ah PBB Components

Component	Description
BCB	Backbone core bridge
BEB	Backbone edge bridge
CE	Customer equipment

Component	Description
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:

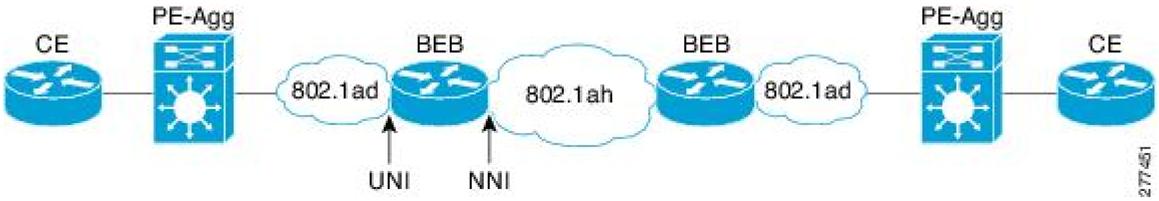
- S-Tagged Service:
    - In multiplexed environments each S-tag maps to an I-SID and may be retained or removed.
    - In bundled environments multiple S-tags map to the same I-SID and the S-tags must be retained.
- C-Tagged Service:
    - In multiplexed environments each C-tag maps to an I-SID and 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 3: IEEE 802.1ah L2 Bridging Network



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

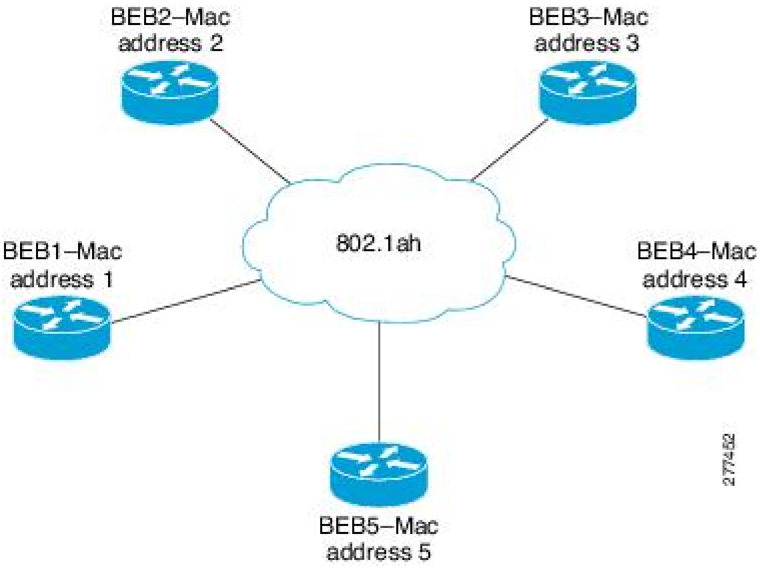
Table 19: L2 Bridging Network Components

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

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 4: 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.

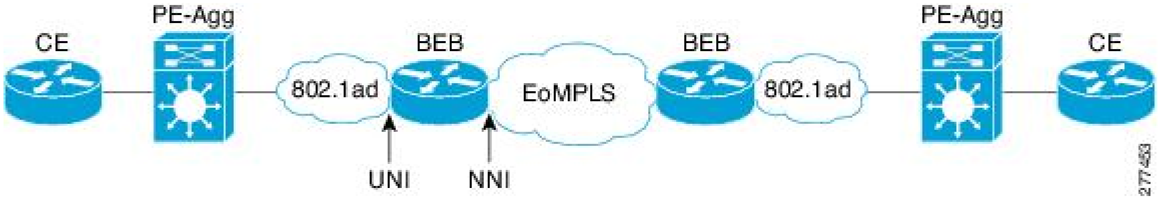
All packets within the 802.1ah network must be sent to a specified MAC address. The address is a static entry in the MAC address tables in the backbone core bridges. If a default MAC tunnel address is not specified in the table, then all unknown unicast packets and customer multicast traffic are sent 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 5: 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.

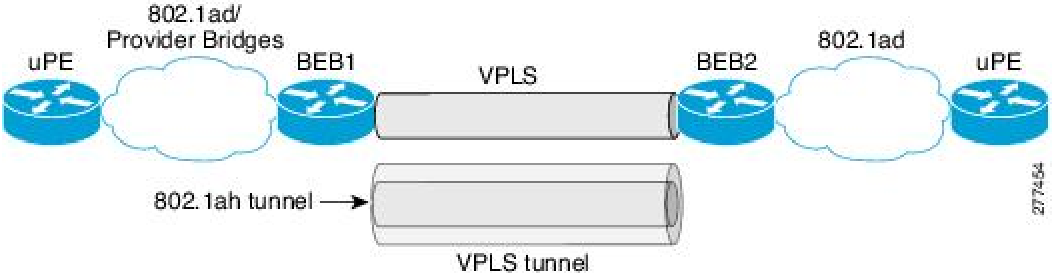
IEEE 802.1ah for Virtual Private LAN Services

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 6: 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.
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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.

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.

	Command or Action	Purpose
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.
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.

	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 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.

	Command or Action	Purpose
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.
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.
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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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 20: 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**.

CHAPTER 14

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 caveats and feature information, see Bug Search Tool and 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 module.

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 operation, administration, and management (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 Local Management Interface (LMI) relies on Ethernet connectivity fault management (CFM) for the status of an Ethernet virtual circuit (EVC), the remote user network interface (UNI) identifier associated with an EVC, and remote UNI status.
- Ethernet LMI customer edge (CE) is available only on routing ports on routing platforms. For information about Ethernet LMI provider edge (PE) functionality on switching platforms, see the “Configuring Ethernet CFM and E-LMI” chapter of the *Cisco ME 3400 Switch Software Configuration Guide*.
- Not all Cisco software releases support autoconfiguration of CE devices.

Information About Enabling Ethernet Local Management Interface

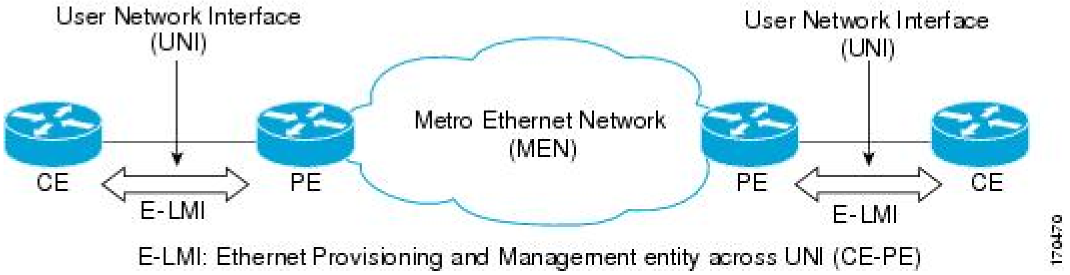
EVC

An Ethernet virtual circuit (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 customer edge (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 ATM.

Ethernet LMI

Ethernet Local Management Interface (LMI) is an Ethernet layer operation, administration, and management (OAM) protocol between a customer edge (CE) device and the provider edge (PE) device 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 virtual circuits (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 user network identifier (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

- 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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet lmi global**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet lmi global** **Example:** `Device(config)# ethernet lmi global`	Enables Ethernet Local Management Interface (LMI) on all supported interfaces on the device.

Step 4	**end**
	Example:
	`Device# end`
		Returns to privileged EXEC mode.

Enabling Ethernet LMI on a Single 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/0`	Specifies an interface and enters interface configuration mode.
Step 4	**ethernet lmi interface** **Example:** `Device(config-if)# ethernet lmi interface`	Enables Ethernet Local Management Interface (LMI) on the interface.

Step 5	**end**
	Example:
	`Device# end`
		Returns 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

```
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 for Enabling Ethernet Local Management Interface

Related Documents

Related Topic	Document Title
Ethernet Connectivity Fault Management (CFM)	"Configuring Ethernet Connectivity Fault Management in a Service Provider Network" in the *Cisco IOS Carrier Ethernet Configuration Guide*
Configuring CFM and Ethernet Local Management Interface (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 *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*

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 21: 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 15.3(1)S	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 introduced in Cisco IOS Release 12.4(9)T. This feature was implemented on the Cisco 7600 router in Cisco IOS Release 12.2(33)SRB. This feature was integrated into Cisco IOS Release 15.3(1)S. 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.

CHAPTER 15

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 caveats and feature information, see Bug Search Tool and 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 module.

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

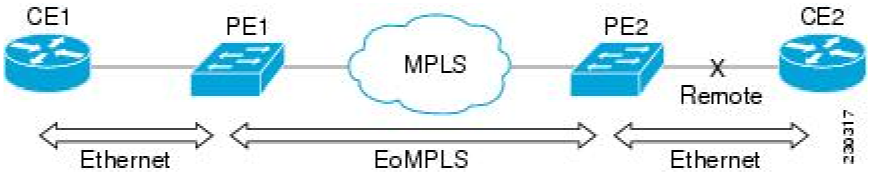
The OAM manager is an infrastructure element that streamlines interaction between OAM protocols. The OAM manager requires two interworking OAM protocols, in this case Ethernet LMI and MPLS LDP.

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**}	Configures either CFM or LDP as an OAM protocol.

	Command or Action	Purpose
	Example: `Router(config-evc)# oam protocol ldp`	- In this example, LDP is the protocol being configured.

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

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 22: 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**.

CHAPTER 16

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 Operation, Administration, and Management (OAM) requirements on Ethernet's traditional operations, which had 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.

The “Configuring Ethernet Local Management Interface at a Provide Edge” module provides general information about configuring an 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 caveats and feature information, see Bug Search Tool and 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 module.

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 Operation, Administration, and Management (OAM) must be operational in the network.
- For Ethernet OAM to operate, the provider edge (PE) side of a connection must be running Ethernet Connectivity Fault Management (CFM) and Ethernet Local Management Interface (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 nonstop 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 Local Management Interface (LMI) is not supported on routed ports, EtherChannel port channels, ports that belong to an EtherChannel, private VLAN ports, IEEE 802.1Q tunnel ports, Ethernet over Multiprotocol Label Switching (MPLS) ports, or Ethernet Flow Points (EFPs) on trunk 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 customer edge (CE) device.

Information About Configuring Ethernet Local Management Interface at a Provider Edge

Ethernet Virtual Circuits Overview

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 customer edge (CE) device 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 another alternative service such as ATM.

Ethernet LMI Overview

Ethernet Local Management Interface (LMI) is an Ethernet Operation, Administration, and Management (OAM) protocol between a customer edge (CE) device and a provider edge (PE) device. Ethernet LMI provides CE devices with the status of Ethernet virtual circuits (EVCs) for large Ethernet metropolitan-area networks (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 Connectivity Fault Management (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 Overview

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

OAM Manager Overview

The OAM manager is an infrastructure element that streamlines interaction between Operation, Administration, and Management (OAM) protocols. The OAM manager requires two interworking OAM protocols, Ethernet Connectivity Fault Management (CFM) and Ethernet Local Management Interface (LMI). No interactions are required between Ethernet LMI and the OAM manager on the customer edge (CE) side. On the User 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 the 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 Ethernet virtual circuit (EVC) status.

The OAM manager calculates EVC status given the number of active user network interfaces (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 is as follows:

- 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 Ethernet virtual circuit (EVC) to the customer edge (CE) device
- Communication of EVC and user network interface (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, high availability (HA) is a requirement, especially on Ethernet operations, administration, and management (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 customer edge (CE), provider edge (PE), and access aggregation PE (uPE) network nodes based on information received by protocols such as Ethernet Local Management Interface (LMI), Connectivity Fault Managment (CFM), and 802.3ah. This status information is used to either stop traffic or switch to backup paths when an EVC is down.

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 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

- Elimination of network downtime for Cisco 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 the system delivers higher service levels due to the elimination of network downtime during upgrades

NSF SSO Support in Ethernet LMI

The redundancy configurations stateful switchover (SSO) and nonstop forwarding (NSF) are supported in Ethernet Local Management Interface (LMI) and are automatically enabled. A switchover from an active to a standby Route Processor (RP) or a standby Route Switch Processor (RSP) occurs when the active RP or RSP 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 or RSP switchover. NSF also interoperates with the SSO feature to minimize network downtime following a switchover.

For detailed information about the SSO and NSF features, see the *High Availability Configuration Guide*.

ISSU Support in Ethernet LMI

In Service Software Upgrade (ISSU) allows you to perform a Cisco software upgrade or downgrade without disrupting packet flow. Ethernet Local Management Interface (LMI) performs updates of the parameters within the Ethernet LMI database to the standby route processor (RP) or standby route switch processor (RSP). This checkpoint data requires ISSU capability to transform messages from one release to another. All the components that perform active processor to standby processor 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 *High Availability Configuration Guide*.

How to Configure Ethernet Local Management Interface at a Provider Edge

Configuring Ethernet LMI Interaction with CFM

For Ethernet Local Management Interface (LMI) to function with Connectivity Fault Management (CFM), you must configure Ethernet virtual circuits (EVCs), Ethernet service instances including untagged Ethernet flow points (EFPs), and Ethernet LMI customer VLAN mapping. Most of the configuration occurs on the provider edge (PE) device on the interfaces connected to the customer edge (CE) device. On the CE device, you need only enable Ethernet LMI on the connecting interface. Also, you must configure operations, administration, and management (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 user network interface (UNI) service type, Ethernet virtual circuit (EVC), Ethernet service instance, or customer edge (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 provider edge (PE) device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **service** *csi-id* **evc** *evc-name* **vlan** *vlan-id*
5. **continuity-check**
6. **continuity-check interval** *time*
7. **exit**
8. **exit**
9. **ethernet evc** *evc-id*
10. **oam protocol** {**cfm domain** *domain-name* | **ldp**}
11. **uni count** *value* [**multipoint**]
12. **exit**
13. Repeat Steps 3 through 12 to define other CFM domains that you want OAM manager to monitor.
14. **interface** *type number*
15. **service instance** *id* **ethernet** [*evc-id*]
16. **ethernet lmi ce-vlan map** {*vlan-id* [**untagged**] | **any** | **default** | **untagged**}
17. **ethernet lmi interface**
18. **encapsulation dot1q** *vlan-id*
19. **bridge-domain** *domain-number*
20. **cfm mep domain** *domain-name* **mpid** *mpid-id*
21. **exit**
22. **service instance** *service-instance-id* **ethernet**
23. **encapsulation untagged**
24. **l2protocol peer**
25. **bridge-domain** *bridge-domain-number*
26. **exit**
27. **ethernet uni** [**bundle** [**all-to-one**] | **id** *uni-id* | **multiplex**]
28. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain cstmr1 level 3`	Defines a Connectivity Fault Management (CFM) domain, sets the domain leve,l and enters Ethernet CFM configuration mode.
Step 4	**service** *csi-id* **evc** *evc-name* **vlan** *vlan-id* **Example:** `Device(config-ecfm)# service csi2 evc evc_1 vlan 10`	Defines a universally unique customer service instance (CSI) and VLAN ID within the maintenance domain, and enters Ethernet CFM service configuration mode.
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Enables the transmission of continuity check messages (CCMs).
Step 6	**continuity-check interval** *time* **Example:** `Device(config-ecfm-srv)# continuity-check interval 1s/10s/1m/10m`	Enables the transmission of continuity check messages (CCMs) at specific intervals.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet CFM configuration mode.
Step 8	**exit** **Example:** `Device(config-ecfm)# exit`	Returns to global configuration mode.

	Command or Action	Purpose
Step 9	**ethernet evc** *evc-id* **Example:** `Device(config)# ethernet evc 50`	Defines an EVC and enters EVC configuration mode.

Step 10	**oam protocol** {**cfm domain** *domain-name* | **ldp**}
	Example:
	`Device(config-evc)# oam protocol cfm domain cstmr1`
		Configures the Ethernet virtual circuit (EVC) operations, administration, and management (OAM) protocol as CFM 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 11	**uni count** *value* [**multipoint**] **Example:** `Device(config-evc)# uni count 3`	(Optional) Sets the User Network Interface (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 12	**exit** **Example:** `Device(config-evc)# exit`	Returns to global configuration mode.
Step 13	Repeat Steps 3 through 12 to define other CFM domains that you want OAM manager to monitor. **Example:** —	
Step 14	**interface** *type number* **Example:**	Specifies a physical interface connected to the CE device and enters interface configuration mode.
Step 15	**service instance** *id* **ethernet** [*evc-id*] **Example:** `Device(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.

	Command or Action	Purpose

Step 16	**ethernet lmi ce-vlan map** {*vlan-id* [**untagged**] | **any** | **default** | **untagged**}
	Example:
	`Device(config-if-srv)# ethernet lmi ce-vlan map 30`
		Configures an Ethernet LMI customer VLAN-to-EVC map for a particular UNI.
		Note To specify both VLAN IDs and untagged VLANs in the map, specify the VLAN IDs first and then specify the **untagged** keyword as follows: **ethernet lmi ce-vlan map 100,200,300,untagged**. Also, if the **untagged** keyword is not specified in the map configuration, the main interface line protocol on the Customer Edge (CE) device will be down.

Step 17	**ethernet lmi interface** **Example:** `Device(config-if-srv)# ethernet lmi interface`	Enables Ethernet local management interface (LMI) on a UNI.
Step 18	**encapsulation dot1q** *vlan-id* **Example:** `Device(config-if-srv)# encapsulation dot1q 2`	Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance.
Step 19	**bridge-domain** *domain-number* **Example:** `Device(config-if-srv)# brdige-domain 1`	Binds a service instance to a bridge domain instance.
Step 20	**cfm mep domain** *domain-name* **mpid** *mpid-id* **Example:** `Device(config-if-srv)# cfm mep domain provider mpid 10`	Configures a maintenance endpoint (MEP) for a domain.
Step 21	**exit** **Example:** `Device(config-if-srv)# exit`	Returns to interface configuration mode.
Step 22	**service instance** *service-instance-id* **ethernet** **Example:** `Device(config-if)# service instance 22 ethernet`	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.

	Command or Action	Purpose
Step 23	**encapsulation untagged** **Example:** `Device(config-if-srv)# encapsulation untagged`	Defines the matching criteria to map untagged ingress Ethernet frames on an interface to the appropriate service instance.
Step 24	**l2protocol peer** **Example:** `Device(config-if-srv)# l2protocol peer`	Configures transparent Layer 2 protocol peering on the interface.
Step 25	**bridge-domain** *bridge-domain-number* **Example:** `Device(config-if-srv)# bridge-domain 1`	Binds a service instance to a bridge domain instance.
Step 26	**exit** **Example:** `Device(config-if)# exit`	Returns to interface configuration mode.
Step 27	**ethernet uni** [**bundle** [**all-to-one**] | **id** *uni-id* | **multiplex**] **Example:** `Device(config-if)# ethernet uni bundle`	Sets UNI bundling attributes.

Step 28	**end**
	Example:
	`Device(config-if)# end`
		Returns 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 Local Management Interface (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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(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:** `Device(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:
	`Device(config-if)# ethernet lmi n393 10`
		Configures Ethernet LMI parameters for the UNI.

Step 6	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Displaying Ethernet LMI and OAM Manager Information

Perform this task to display Ethernet Local Management Interface (LMI) or Operation, Administration, and Management (OAM) manager information. After step 1, 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:** `Device> 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:
	`Device# show ethernet lmi evc`
		Displays information that was sent to the customer edge (CE).

Step 3	**show ethernet service evc** [**detail** | **id** *evc-id* [**detail**] | **interface** *type number* [**detail**]] **Example:** `Device# show ethernet service evc`	Displays information about all Ethernet virtual circuits (EVCs) or about a specified EVC.
Step 4	**show ethernet service instance** [**detail** | **id** *id* | **interface** *type number* | **policy-map** | **stats**] **Example:** `Device# show ethernet service instance detail`	Displays information about customer service instances.

	Command or Action	Purpose
Step 5	**show ethernet service interface** [*type number*] [**detail**] **Example:** `Device# 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.

Examples

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

```
Device# show ethernet lmi evc

St  EVC Id                                                           Port
--- ------------------------------------------------------------ --------------
A  EVC_MP2MP_101                                                Gi0/1
A  EVC_P2P_110                                                  Gi0/1
```

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

```
Device# show ethernet service evc

Identifier                     Type  Act-UNI-cnt Status
50                             MP-MP     0       NotDefined
```

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

```
Device#

Interface: Gigabitethernet
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 is sample output from the **show ethernet service instance** command using the **detail** keyword:

```
Device# show ethernet service instance detail

Service Instance ID: 400
Associated Interface: GigabitEthernet
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 Operation, Administration, and Management (OAM) manager, Connectivity Fault Management (CFM), and Ethernet Local Management Interface (LMI) on a provider edge (PE) device. In this example, a bridge domain is specified.

```
Device> enable
Device# configure terminal
Device(config)# ethernet cfm global
Device(config)# ethernet cfm domain provider level 4
Device(config-ecfm)# service customer_1 evc test1 vlan 10
Device(config-ecfm-srv)# continuity-check
Device(config-ecfm-srv)# continuity-check interval 1s/10s/1m/10m
Device(config-ecfm-srv)# exit
Device(config-ecfm)# exit
Device(config)# ethernet evc test1
Device(config-evc)# uni count 3
Device(config-evc)# oam protocol cfm domain provider
Device(config-evc)# exit
Device(config)#
Device(config-if)# ethernet lmi interface
Device(config-if)# ethernet uni id CISCO
Device(config-if)# service instance 1 ethernet
Device(config-if-srv)# encapsulation untagged
Device(config-if-srv)# l2protocol peer
Device(config-if-srv)# bridge-domain 1
Device(config-if-srv)# exit
Device(config-if)# service instance 2 ethernet1
Device(config-if-srv)# ethernet lmi ce-vlan map 101
Device(config-if-srv)# encapsulation dot1q 2
Device(config-if-srv)# bridge-domain 2
Device(config-if-srv)# cfm mep domain provider mpid 10
Device(config-if-srv-ecfm-mep)# end
```

This example shows a configuration of OAM manager, CFM, and Ethernet LMI over an Xconnect configuration:

```
Device> enable
Device# configure terminal
Device(config)# ethernet cfm global
Device(config)# ethernet cfm domain provider level 4
Device(config-ecfm)# service customer_1 evc test1
Device(config-ecfm-srv)# continuity-check
Device(config-ecfm-srv)# continuity-check interval 1s,10s,1m,10m
Device(config-ecfm-srv)# exit
Device(config-ecfm)# exit
Device(config)# ethernet evc test1
Device(config-evc)# oam protocol cfm domain provider
Device(config-evc)# exit
Device(config)#
Device(config-if)# ethernet lmi interface
Device(config-if)# ethernet uni id CISCO
Device(config-if)# service instance 1 ethernet
Device(config-if-srv)# encapsulation untagged
Device(config-if-srv)# l2protocol peer
Device(config-if-srv)# bridge-domain 1
Device(config-if-srv)# exit
Device(config-if)# service instance 2 ethernet
Device(config-if-srv)# ethernet lmi ce-vlan map 101
Device(config-if-srv)# encapsulation dot1q 2
Device(config-if-srv)# xconnect 10.1.1.1 100 encapsulation mpls
Device(cfg-if-ether-vc-xconn)# exit
Device(config-if-srv)# cfm mep domain provider mpid 10
Device(config-if-srv-ecfm-mep)# end
```

Example: Ethernet LMI on a CE Device Configuration

This example shows how to configure Ethernet Local Management Interface (LMI) globally on a customer edge (CE) device:

```
Device# configure terminal
Device(config)# ethernet lmi global
Device(config)# ethernet lmi ce
Device(config)# exit
```

Additional References for Configuring Ethernet Local Management Interface at a Provider Edge

Related Documents

Related Topic	Document Title
Ethernet Connectivity Fault Management (CFM)	“Configuring Ethernet Connectivity Fault Management in a Service Provider Network” in the *Carrier Ethernet Configuration Guide*
Ethernet Local Management Interface (LMI)	“Enabling Ethernet Local Management Interface” in the *Carrier Ethernet Configuration Guide*
Remote Port Shutdown feature	“Configuring Remote Port Shutdown” in the *Carrier Ethernet Configuration Guide*
IEEE 802.3ah	*IEEE 802.3ah Ethernet in the First Mile*
Cisco high availability (HA) configuration information	*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)*

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 23: 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**.

Feature Name	Releases	Feature Information
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**.
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**.

CHAPTER 17

Configuring IEEE 802.3ad Link Bundling and Load Balancing

This document describes how the IEEE 802.3ad link bundling and load balancing leverages the EtherChannel infrastructure within Cisco 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 caveats and feature information, see Bug Search Tool and 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 module.

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.
- 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.
- 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.
- The adaptive algorithm does not apply to service control engines (SCEs) when EtherChannel load distribution is used.

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 uses the following parameters:

- 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.
-

Benefits of IEEE 802.3ad Link Bundling

- Increased network capacity without changing physical connections or upgrading hardware
- Cost savings from the 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.

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. The maximum number of ports that can be bundled varies by platform. 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

In earlier Cisco software releases, 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 EtherChannel Load Distribution 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:

```
Device(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:

```
Device# 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

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

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface port-channel** *channel-number* **Example:** `Device(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:
	`Device(config-if)# channel-group 25 mode active`
		Configures the interface in a channel group and sets it as active.
		• In active mode, the port initiates negotiations with other ports by sending Link Aggregate Control Protocol (LACP) packets.

	Command or Action	Purpose

Step 5	**end**
	Example:
	`Device(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*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface port-channel** *channel-number* **Example:** `Device(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:** `Device(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:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Step 6	**show running-config interface port-channel** *group-number* **Example:** `Device# show running-config interface port-channel 10`	Displays the port channel configuration.

Example

This example shows how to verify the configuration:

```
Device# 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

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface port-channel** *channel -number* **Example:** `Device(config)# interface port-channel 5`	Creates a port channel and enters interface configuration mode.

Step 4	**channel-group** *channel-group-number* **mode** {**active** | **passive**}
	Example:
	`Device(config-if)# channel-group 5 mode active`
		Includes the interface as part of the port channel bundle.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Setting LACP System Priority

Perform this task to set the Link Aggregation Control Protocol (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**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**lacp system-priority** *priority* **Example:** `Device(config)# lacp system-priority 200`	Sets the system priority.

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Step 5	**show lacp sys-id** **Example:** `Device# show lacp sys-id`	Displays the system ID, which is a combination of the system priority and the MAC address of the device.

Example

This example shows how to verify the LACP configuration:

```
Device# show lacp sys-id
20369,01b2.05ab.ccd0
```

Adding and Removing Interfaces from a 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(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:
	`Device(config-if)# channel-group 5 mode active`
		Adds an interface to a channel group.

Step 5	**no channel-group** **Example:** `Device(config-if)# no channel-group`	Removes the interface from the channel group.

	Command or Action	Purpose

Step 6	**end**
	Example:
	`Device(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 a Link Aggregate Control Protocol (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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface port-channel 1`	Creates a port-channel virtual interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**lacp min-bundle** *min-bundle* **Example:** `Device(config-if)# lacp min-bundle 5`	Sets the minimum threshold of active links.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Monitoring LACP Status

SUMMARY STEPS

1. **enable**
2. **show lacp** {*number* \| **counters** \| **internal** \| **neighbor** \| **sys-id**}

DETAILED STEPS

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

Step 2	**show lacp** {*number* | **counters** | **internal** | **neighbor** | **sys-id**}
	Example:
	`Device# show lacp internal`
		Displays internal device information.

Troubleshooting Tips

Use the **debug lacp** command to display Link Aggregate Control Protocol (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:

```
Device# debug lacp all
Link Aggregation Control Protocol all debugging is on
Device#
*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:

```
Device#
*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 Link Aggregate Control Protocol (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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface port-channel 1`	Creates a port-channel virtual interface and enters interface configuration mode.
Step 4	**lacp direct-loadswap** **Example:** `Device(config-if)# lacp direct-loadswap`	Enables LACP single fault direct load balancing.

Step 5	**end**
	Example:
	`Device(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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(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:
	`Device(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:
	`Device(config)# end`
		Returns to privileged EXEC mode.

Enabling 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(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:
	`Device(config-if)# port-channel load-balance weighted link all`
		Configures weighted load balancing on port-channel member links.

Step 5	**end**
	Example:
	`Device(config-if)# end`
		Returns to privileged EXEC mode.

Configuration Examples for 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:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface port-channel5
Device(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
Device(config-if)#
Device(config-if)# channel-group 5 mode active
Device(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
Device(config-if)# end
Device#
*Aug 20 17:08:00.933: %SYS-5-CONFIG_I: Configured from console by console
Device# 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
Device# 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:

```
Device# 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
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface gigabitethernet 5/0/0
Device(config-if)# channel-group 5 mode active
Device(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
Device(config-if)# end
Device#
*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

Device# 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
Device# 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:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface gigabitethernet 7/0/0
Device(config-if)# no channel-group
Device(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
Device(config-if)# end
Device#
*Aug 20 17:15:58.209: %SYS-5-CONFIG_I: Configured from console by console
Device#
*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

Device#
*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
Device# 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 Link Aggregation Protocol (LACP) activity that you can monitor by using the **show lacp** command.

```
Device# 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
Device# 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
Device# 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
Device# 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
Device# 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
Device# show 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.

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

```
Device(config)# interface GigabitEthernet5/0/2
Device(config-if)# channel-group 10 mode on link 1
Device(config-if)# exit
Device(config)# interface GigabitEthernet5/0/3
Device(config-if)# channel-group 10 mode on link 2
Device(config-if)# exit
Device(config)# interface GigabitEthernet5/0/4
Device(config-if)# channel-group 10 mode on link 3
Device(config-if)# exit
!
Device(config)# interface Port-channel10
Device(config-if)# port-channel load-balance link 1
Device(config-if)# service-instance 100-150
Device(config-if)# port-channel load-balance weighted link 2,3
Device(config-if)# port-channel load-balance weighted default weight 2
Device(config-if)# port-channel load-balance weighted rebalance disable
Device(config-if)# service instance 100 ethernet
Device(config-if-srv)# encapsulation dot1q 100
Device(config-if-srv)# exit
Device(config-if)# service instance 101 ethernet
Device(config-if-srv)# encapsulation dot1q 101
Device(config-if-srv)# exit
Device(config-if)# service instance 200 ethernet
Device(config-if-srv)# encapsulation dot1q 200
Device(config-if-srv)# exit
Device(config-if)# service instance 201 ethernet
Device(config-if-srv)# encapsulation dot1q 201
Device(config-if-srv)# exit
Device(config-if)# service instance 300 ethernet
Device(config-if-srv)# encapsulation dot1q 300
Device(config-if-srv)# weight 5
Device(config-if-srv)# end
```

Additional References for IEEE 802.3ad Link Bundling and Load Balancing

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 the Cisco Catalyst 3850 Series Switch	*Catalyst 3850 Series Switch Configuration Guide*
Configuring Carrier Ethernet	*Carrier Ethernet Configuration Guide*
Link Aggregation Control Protocol (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 Command 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

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

Feature Name	Releases	Feature Information
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**.
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**.

Feature Name	Releases	Feature Information
IEEE 802.3ad Link Aggregation (LACP)	12.2(31)SB2 12.2(33)SRB 12.2(33)SRC 15.0(1)S	The IEEE 802.3ad Link Aggregation 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 third-party devices. 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**.
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.

CHAPTER 18

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 caveats and feature information, see Bug Search Tool and 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 module.

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.
- 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

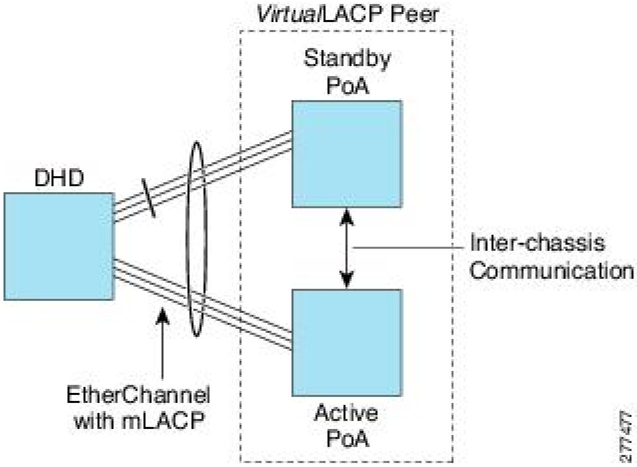
- mLACP does not support Fast Ethernet.
- mLACP does not support half-duplex links.
- mLACP does not support multiple neighbors.
- Converting a port channel to mLACP can cause a service disruption.
- The maximum number of member links per LAG per PoA is restricted by the maximum number of ports per port 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):

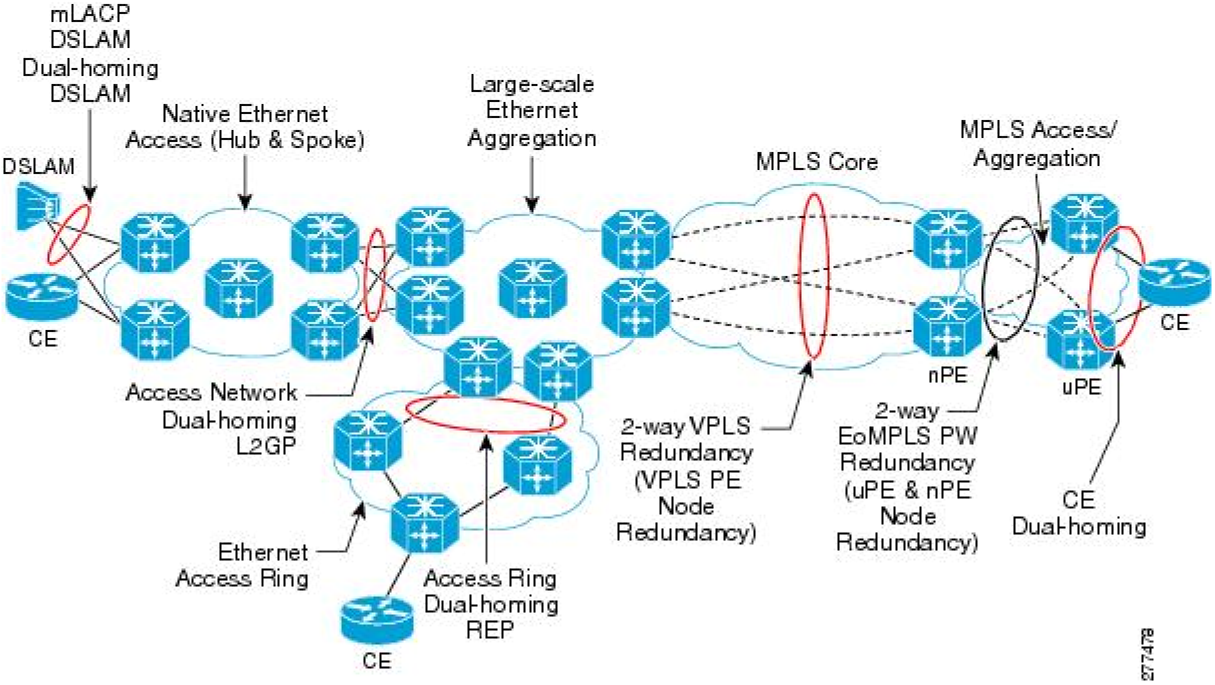
- Attachment circuit active or standby status selection--This selection can be performed by the access node or network, the aggregation node, or combination of the two. For mLACP, the attachment circuit status selection is determined through collaboration between the DHD and the PoAs.
- Pseudowire forwarding 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.

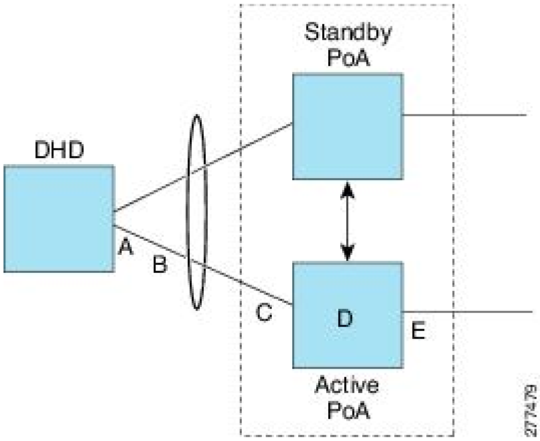
Port Key

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

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

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.

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:

```
Device# 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 25: 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.

Adjacency Type	Adjacency Status	Meaning
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.

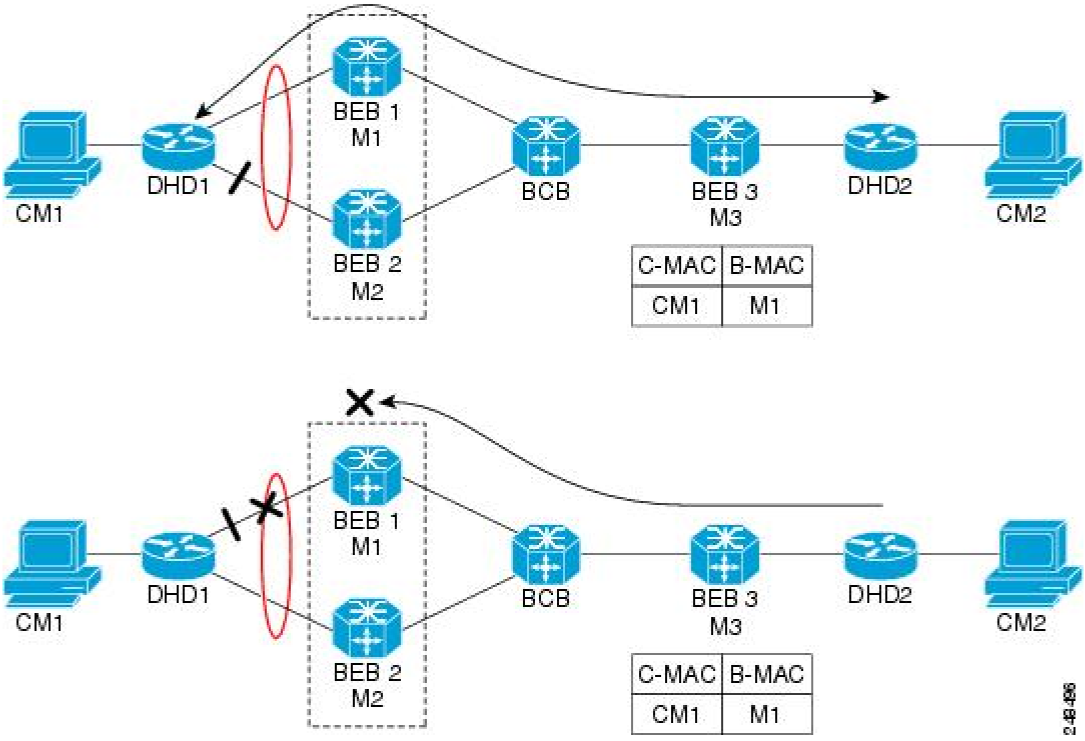
At the failover from a primary PoA to a secondary PoA, a service experiences traffic blackholing when the DHD in question remains inactive and while other remote devices in the network are attempting to send traffic to that DHD. Remote bridges in the network have stale MAC entries pointing to the failed 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

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.



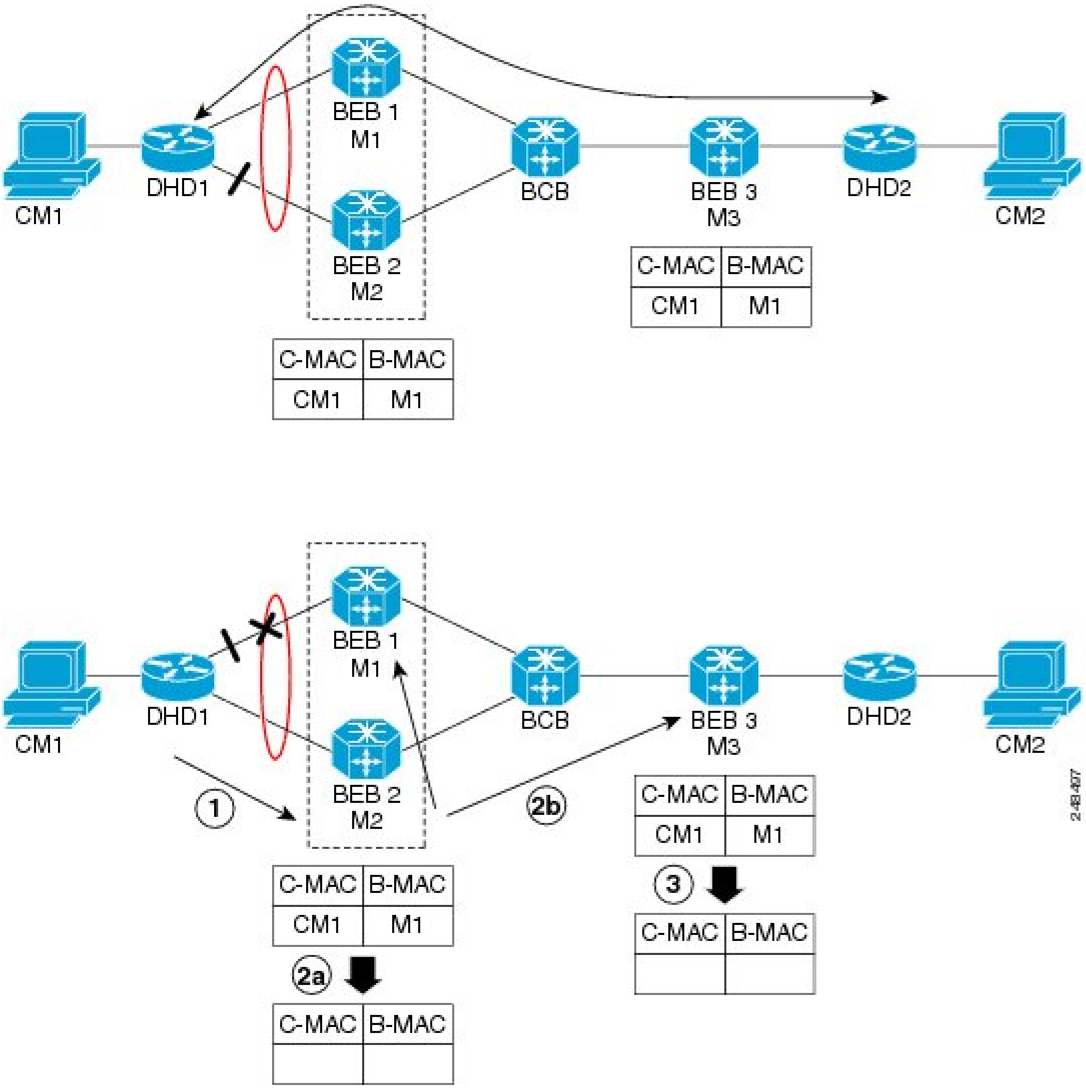
Device DHD1 is dual-homed to two 802.1ah backbone edge bridges (BEB1 and BEB2). Assume that initially the primary path is through BEB1. In this configuration BEB3 learns that the host behind DHD1 (with MAC address CM1) is reachable via the destination B-MAC M1. If the link between DHD1 and BEB1 fails 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.
Step 3	**redundancy** **Example:** `Router(config)# redundancy`	Enters redundancy configuration mode.

	Command or Action	Purpose
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.
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, on page 440.

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. **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.

	Command or Action	Purpose
		• 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.
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	**end**
	Example:
	`Router(config-r-ic)# 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.
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.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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 VPWS on ME3600 Series Switches

Perform this task to provide Layer 2 VPN service redundancy for VPWS on Cisco ME3600, ME3600X 24CX, ME3800 series switches.

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. **switchport mode trunk**
9. **switchport trunk allowed vlan none**
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.

	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	**switchport mode trunk** **Example:** `Router(config-if)# switchport mode trunk`	Specifies the port channel as trunking VLAN Layer 2 interface.
Step 9	**switchport trunk allowed vlan none** **Example:** `Router(config-if)# switchport trunk allowed vlan none`	Sets the list of allowed VLANs that transmit traffic from this interface in tagged format when in trunking mode.
Step 10	**lacp fast-switchover** **Example:** `Router(config-if)# lacp fast-switchover`	Enables LACP 1-to-1 link redundancy.

	Command or Action	Purpose
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.
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.

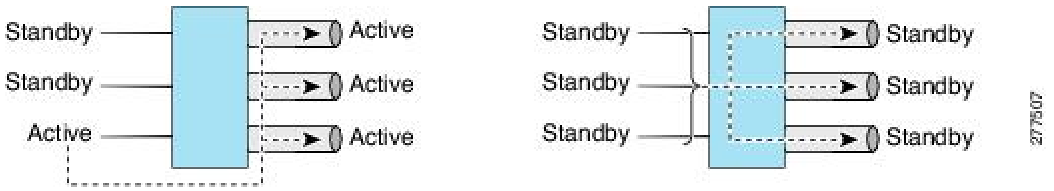
	Command or Action	Purpose
Step 19	**encapsulation dot1q** *vlan-id* [**,** *vlan-id*[**-** *vlan-id*]] [**native**] **Example:** `Router(config-if-srv)# encapsulation dot1q 100`	Enables IEEE 802.1Q encapsulation of traffic on a specified subinterface in a VLAN.
Step 20	**exit** **Example:** `Router(config-if-srv)# 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 ether-pw`	Binds an attachment circuit to a pseudowire.
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.

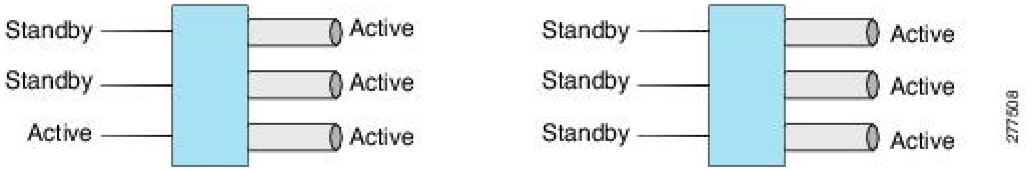
Configuring Redundancy for VPLS

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.
Step 10	**lacp fast-switchover** **Example:** `Router(config-if)# lacp fast-switchover`	Enables LACP 1-to-1 link redundancy.

	Command or Action	Purpose
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.
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.

	Command or Action	Purpose
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.
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.

Steps for Configuring Redundancy for VPLS on ME3600 Series Switches

Perform the following task to configure redundancy for VPLS on Cisco ME3600, ME3600X 24CX, ME3800 series switches.

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. **switchport mode trunk**
10. **switchport trunk allowed vlan none**
11. **lacp fast-switchover**
12. **lacp max-bundle** *max-bundles*
13. **exit**
14. **redundancy**
15. **interchassis group** *group-id*
16. **exit**
17. **exit**
18. **interface port-channel** *port-channel- number*
19. **service instance** *id* **ethernet** [*evc-name*]
20. **encapsulation dot1q** *vlan-id* [**,** *vlan-id*[**-** *vlan-id*]] [**native**]
21. **bridge-domain** *bridge-id* [**split-horizon** [**group** *group-id*]]
22. **exit**
23. **interface vlan** *vlanid*
24. **no ip address**
25. **xconnect vfi** *vfi-name*
26. **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	**switchport mode trunk** **Example:** `Router(config-if)# switchport mode trunk`	Specifies the port channel as trunking VLAN Layer 2 interface.
Step 10	**switchport trunk allowed vlan none** **Example:** `Router(config-if)# switchport trunk allowed vlan none`	Sets the list of allowed VLANs that transmit traffic from this interface in tagged format when in trunking mode.

	Command or Action	Purpose
Step 11	**lacp fast-switchover** **Example:** `Router(config-if)# lacp fast-switchover`	Enables LACP 1-to-1 link redundancy.
Step 12	**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 13	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 14	**redundancy** **Example:** `Router(config)# redundancy`	• Enters redundancy configuration mode.
Step 15	**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 16	**exit** **Example:** `Router(config-r-ic)# exit`	Exits interchassis redundancy mode.
Step 17	**exit** **Example:** `Router(config-red)# exit`	Exits redundancy configuration mode.
Step 18	**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 19	**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 20	**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 21	**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 22	**exit** **Example:** `Router(config-if-srv)# exit`	Exits service instance configuration mode.
Step 23	**interface vlan** *vlanid* **Example:** `Router(config-if)# interface vlan 200`	Creates a dynamic switch virtual interface (SVI).
Step 24	**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 25	**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 26	**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.

	Command or Action	Purpose
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.
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 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.
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.

	Command or Action	Purpose
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.
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.

Configuring Hierarchical VPLS on ME3600 Series Switches

Perform this task to configure Hierarchical VPLS (H-VPLS) on Cisco ME3600, ME3600X 24CX, ME3800 series switches.

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. **switchport mode trunk**
10. **switchport trunk allowed vlan none**
11. **lacp fast-switchover**
12. **lacp max-bundle** *max-bundles*
13. **exit**
14. **redundancy**
15. **interchassis group** *group-id*
16. **exit**
17. **exit**
18. **interface port-channel** *port-channel- number*
19. **service instance** *id* **ethernet** [*evc-name*]
20. **encapsulation dot1q** *vlan-id* [**,** *vlan-id*[**-** *vlan-id*]] [**native**]
21. **exit**
22. **xconnect** *peer-ip-address* *vc-id* {**encapsulation mpls** | **pw-class** *pw-class-name*} [**pw-class** *pw-class-name*] [**sequencing** {**transmit** | **receive** | **both**}]
23. **backup peer** *peer-router-ip-addr* *vcid* [**pw-class** *pw-class-name*] [**priority** *value*]
24. **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 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.
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	**switchport mode trunk** **Example:** `Router(config-if)# switchport mode trunk`	Specifies the port channel as trunking VLAN Layer 2 interface.
Step 10	**switchport trunk allowed vlan none** **Example:** `Router(config-if)# switchport trunk allowed vlan none`	Sets the list of allowed VLANs that transmit traffic from this interface in tagged format when in trunking mode.

	Command or Action	Purpose
Step 11	**lacp fast-switchover** **Example:** `Router(config-if)# lacp fast-switchover`	Enables LACP 1-to-1 link redundancy.
Step 12	**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 13	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 14	**redundancy** **Example:** `Router(config)# redundancy`	Enters redundancy configuration mode.
Step 15	**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 16	**exit** **Example:** `Router(config-r-ic)# exit`	Exits interchassis redundancy mode.
Step 17	**exit** **Example:** `Router(config-red)# exit`	Exits redundancy configuration mode.
Step 18	**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 19	**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 20	**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 21	**exit** **Example:** `Router(config-if-srv)# exit`	Exits service instance configuration mode.
Step 22	**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.
Step 23	**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 24	**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.

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.

	Command or Action	Purpose
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.

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.

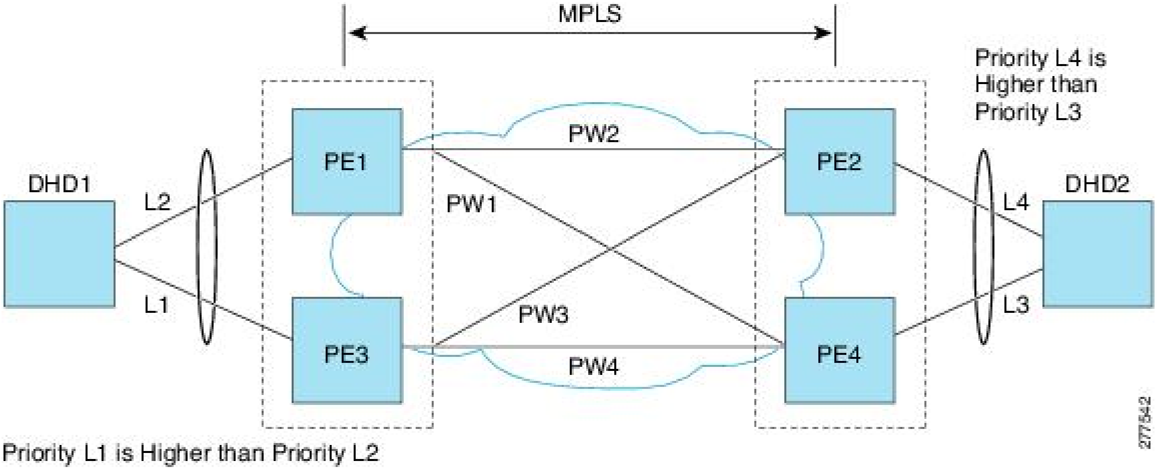
	Command or Action	Purpose
Step 2	**debug mpls l2transport checkpoint** **Example:** `Router# debug mpls l2transport checkpoint`	Enables the display of AToM events when AToM is configured for nonstop forwarding/stateful switchover (NSF/SSO) and Graceful Restart.
Step 3	**debug acircuit checkpoint** **Example:** `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.


MPLS
Priority L4 is Higher than Priority L3
PE1
PW2
PE2
DHD1
L2
PW1
L4
DHD2
L1
PW3
L3
PE3
PW4
PE4
277542
Priority L1 is Higher than Priority L2

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:

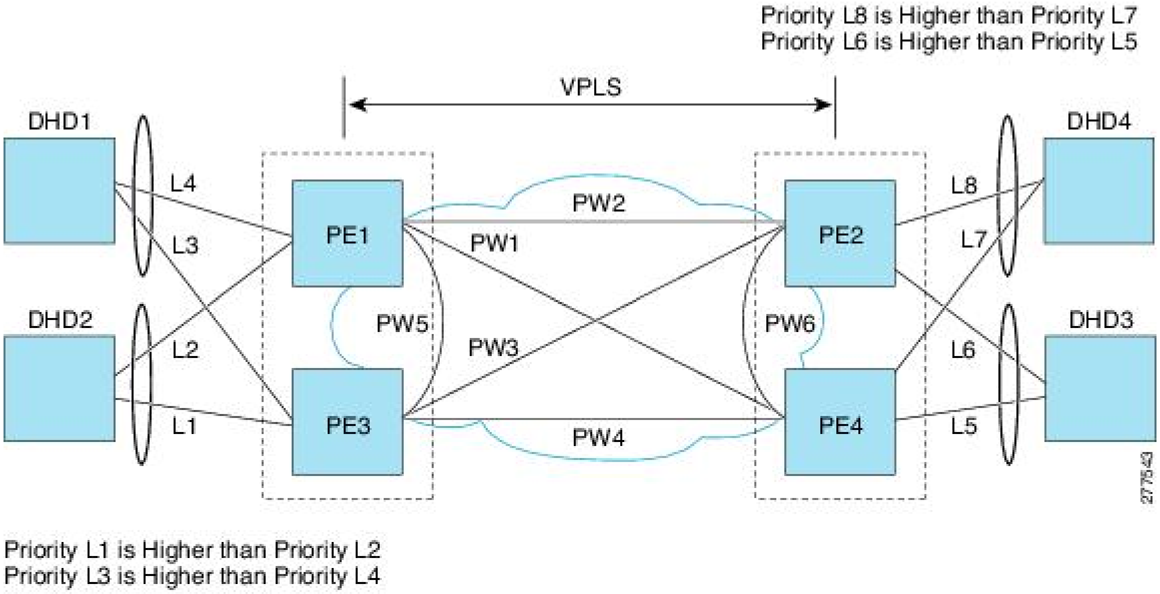
```
mpls ldp graceful-restart
mpls label protocol ldp
mpls ldp graceful-restart
mpls label protocol ldp
!
Redundancy
 mode sso
 interchassis group 1
  member ip 200.0.0.1
  backbone interface Ethernet0/2
  backbone interface Ethernet1/2
  backbone interface Ethernet1/3
  monitor peer bfd
  mlacp node-id 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.


Priority L8 is Higher than Priority L7
Priority L6 is Higher than Priority L5
VPLS
DHD1
DHD2
DHD3
DHD4
PE1
PE2
PE3
PE4
L1
L2
L3
L4
L5
L6
L7
L8
PW1
PW2
PW3
PW4
PW5
PW6
277543
Priority L1 is Higher than Priority L2
Priority L3 is Higher than Priority L4

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:

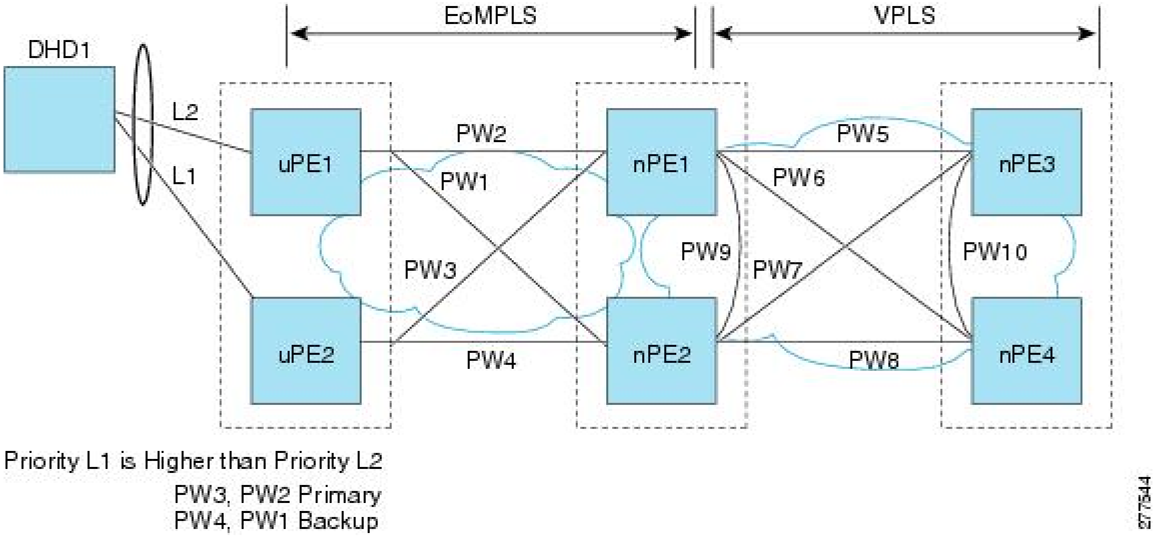
```
mpls ldp graceful-restart
mpls label protocol ldp
!
redundancy
 interchassis group 1
  member ip 200.0.0.1
  backbone interface Ethernet0/2
  monitor peer bfd
  mlacp node-id 1
!
l2 vfi VPLS1 manual
 vpn id 10
 neighbor 210.0.0.1 encapsulation mpls
 neighbor 211.0.0.1 encapsulation mpls
 neighbor 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

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

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

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

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

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 26: 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.

CHAPTER 19

ICCP Multichassis VLAN Redundancy

Carrier Ethernet network high availability can be achieved by employing intra- and inter-chassis redundancy mechanisms. The Multichassis Link Aggregation Control Protocol (mLACP) solution addresses the latter, where a carrier wants dual-homed device (DHD) to two upstream points of attachment (PoA) for redundancy. Some carriers do not run loop prevention control protocols in their access networks, so an alternate redundancy scheme is necessary.

The implementation of mLACP supports DHD with an active/standby topology. Interchassis Communication Protocol (ICCP) Multichassis VLAN Redundancy, also known as Pseudo mLACP, provides a flexible dual-homing redundancy mechanism. It uses similar principles as mLACP. The Pseudo mLACP solution extends the mLACP functionality to support active/active PoAs deployments. This enables flexibility in network planning and efficient resource utilization.

Pseudo mLACP has the following advantages over mLACP:

- Pseudo mLACP supports per-VLAN active/active redundancy without any load-balancing requirements on the CE.
- Pseudo mLACP is independent of the access redundancy mechanism; therefore, it provides a network-based redundancy solution. It allows maximum flexibility for the Provider Edge (PE)-Customer Edge (CE) interoperability in terms of dual-homing redundancy and recovery.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 ICCP Multichassis VLAN Redundancy

- mLACP support is required for Pseudo mLACP.

Restrictions for ICCP Multichassis VLAN Redundancy

- Max bundle should not be configured on a Pseudo mLACP enabled port channel.
- Pseudo mLACP does not work with most of the Layer 2 control protocols or Spanning Tree Protocol (STP) including Multiple Spanning Tree Protocol (MSTP) or VLAN Trunking Protocol (VTP).
- When a service instance is configured under a Pseudo mLACP port channel, all the outer tag VLANs of a service instance must be a part of either a primary VLAN list or a secondary VLAN list.
- Outer VLANs of one service instance cannot be mixed with the primary and secondary VLAN list on a Pseudo mLACP port channel.
- Brute-Force mode configuration is not supported.
- VLAN force-switchover configuration is applicable only for nonrevertive mode.
- The DHD nodes must support the LACP functionality.
- The DHD nodes must support MVRP MAC flush functionality in Pseudo mLACP topology.

Information About ICCP Multichassis VLAN Redundancy

Pseudo mLACP Multihoming Redundancy

The provider edge (PE) ports are configured in such a way that they act as if connected to a virtual device over a Multichassis link aggregation group (MC-LAG) with mLACP. Points of Attachment (PoAs) can be placed in active/active mode with manual VLAN load balancing. DHD ports are configured into two individual port channels that are physically connected to each of the PoAs. Interchassis Communication Protocol (ICCP), with new extensions is used for interchassis communication to control the failover process. Multiple VLAN Registration Protocol (MVRP) lite is used for active VLAN notification and MAC flushing toward the access side. For MAC flushing notification toward the core, MVRP lite, Multiple I-SID Registration Protocol (MIRP) lite, or LDP MAC withdraw can be used.

Pseudo mLACP provides:

- The active/active mode redundancy of two PoAs in a redundancy group. This provides higher bandwidth utilization than mLACP and other active/standby link-level schemes. Pseudo mLACP eliminates the required wasted link bandwidth on the standby PoA.
- Flexible access network topologies, that is, access network dual-homing and access device dual-homing.
- Service provider control over the provisioning, role assignment, failover, and load sharing between PoAs.
- PE node redundancy for Virtual Private Wire Service (VPWS), Virtual Private LAN Service (VPLS), and native Ethernet aggregation.

The DHD is configured with two different port channels that are connected to a single multichassis LAG (mLAG) on the PoA side. The LACP module on the PoAs receives two different port keys from the two different port channels on the DHD. The mLAG on the PoA ignores the port keys from the DHD's LACP PDUs to form a single bundle on the PoA side.

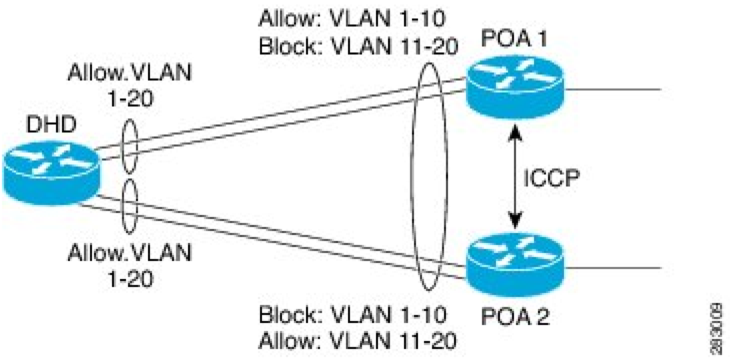
The mLACP module provides failover and recovery notifications to Pseudo mLACP. Reversion delay is processed by the mLACP module. mLACP provides a CLI interface for Pseudo mLACP VLANs and mode configuration. mLACP supports VLAN-based active/active redundancy, in addition to PoA-level active/standby redundancy. VLAN-based active/active redundancy allows you to bundle links on both the PoAs based on the Pseudo mLACP configuration. Pseudo mLACP and mLACP port-channels can be configured together on the same pair of PoAs, and both can use the same redundancy group.

After failover, the new active PoA activates the standby VLAN list on the port-channel. However, to receive traffic on the newly active VLAN's DHD, networks must flush their MAC address table and learn the new MAC address of the new PoA port channel interface. The existing MVRP lite support is used for DHD-side MAC flushing.

Pseudo mLACP Active/Active Support

Pseudo mLACP supports active/active redundancy without the restriction of symmetric VLAN-based load sharing in both the Provider Edge (PE) and the Customer Edge (CE).

Figure 7: Active/Active Support



Pseudo mLACP provides VLAN-based redundancy by allowing you to specify one primary interface and one secondary interface or a PoA pair for each member VLAN. The configuration determines the PoA that will be initially active for a VLAN, by using the primary and secondary VLAN lists under the Pseudo mLACP interface. Only the active PoA will forward frames for the respective VLANs. The standby PoA will be in the blocking mode (bidirectional), dropping all the frames received on the standby VLANs. The failover will occur for all the VLANs in the active/standby list and not on a per-VLAN basis. Pseudo mLACP provides per-port-channel VLAN load balancing. You can statistically configure the primary and secondary VLAN list on each of the PoAs.

The DHD nodes are configured such that each of their uplinks to a PoA operates as an individual port channel. Each interface must be configured to forward all local VLANs on all uplinks belonging to the mLAG.

The data-path forwarding scheme causes the DHD to automatically learn which PoA or interface is active for a given VLAN. This learning occurs at an individual destination MAC address level.

Failure Recovery

Pseudo mLACP uses revertive behavior (which is the default behavior) after the failure recovery to support the active/active model. You can configure a nonrevertive mode.

Reversion occurs the same way that the original failover occurs. The reversion must be initiated by the new active PoA for the given VLANs, by signaling that the PoA is relinquishing its active role for the VLAN. This is done through an ICCP Pseudo mLACP port-state TLV, which indicates that it is no longer in the active mode for the affected VLANs. Upon a TLV receipt, the recovering PoA unblocks the affected VLANs, and triggers MAC flushes toward both the access side and the core side).

mLACP reversion delay applies for Pseudo mLACP operations. However, reversion occurs only for failed-over VLANs. The forced failover mechanism based on dynamic port-priority change cannot be used for Pseudo mLACP because all the member links will remain in the bundle state. Use the **mlacp reversion-delay** command to configure the mLACP reversion timer. Use the **mlacp load-balance force-switchover portchannel** command to configure forced VLAN switchover.

Pseudo mLACP Failover Operations

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

Note mLACP failover will not be triggered if Pseudo mLACP is not configured correctly. If the mLACP failover occurs before the peer PoA is configured with Pseudo mLACP, the failover will occur as long as the peer PoA meets the mLACP failover requirements.

- Access side link or port failure—This failure is triggered by a min-link failure. On receiving a min-link failure, all the active VLANs on the port-channel failover to the other PoA. This failover is initiated by sending a Pseudo mLACP PORT-STATE TLV message, indicating that the port state is DOWN.
- Node failure—The surviving PoA's Pseudo mLACP receives notification of node failure and initiates failover of all VLANs that were in standby mode on all shared mLAGs. After recovery, both POAs synchronize again.
- PoA uplink failure—The failing PoA signals the peer about the core isolation using the Pseudo mLACP PORT-STATE TLV, indicating that the PoA is isolated. It places all the VLANs in the blocking mode.

How to Configure ICCP Multichassis VLAN Redundancy

Configuring a Port Channel for Pseudo mLACP

Perform this task to configure a port channel for Pseudo mLACP.

Before You Begin

Note The redundancy group should be configured. Redundancy group configuration for Pseudo mLACP is the same as for mLACP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface port-channel** *number*
4. **mlacp interchassis group** *group-id*
5. **mlacp mode active-active**
6. **mlacp load-balance primary vlan** *vlan-id*
7. **mlacp load-balance secondary vlan** *vlan-id*
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** *number* **Example:** `Router(config)# interface port-channel 1`	Configures the port channel and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**mlacp interchassis group** *group-id* **Example:** `Router(config-if)# mlacp interchassis group 1`	Specifies that the port channel is an mLACP port-channel .
Step 5	**mlacp mode active-active** **Example:** `Router(config-if)# mlacp mode active-active`	Enables pseudo mLACP operations on the PoA and allows the PoA to form an LACP bundle even if the partner receives an LACP PDU from two different port channels on a dual-homed network (DHN) or dual-homed device (DHD).
Step 6	**mlacp load-balance primary vlan** *vlan-id* **Example:** `Router(config-if)# mlacp load-balance primary vlan 10,20`	Configures the list of primary VLANs that will be active and inactive on the given PoA.
Step 7	**mlacp load-balance secondary vlan** *vlan-id* **Example:** `Router(config-if)# mlacp load-balance secondary vlan 30,100`	Configures the list of secondary VLANs that will be active and inactive on the given PoA.

Step 8	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for ICCP Multichassis VLAN Redundancy

Example: Port Channel Configuration for Pseudo mLACP

The following example shows how to configure the port channel on the active and standby PoA for Pseudo mLACP.

Active PoA-POA1

```
Router# configure terminal
Router(config)# interface port-channel1
Router(config-if)# mlacp interchassis group 1
Router(config-if)# mlacp mode active-active
Router(config-if)# mlacp load-balance primary vlan 10,20
Router(config-if)# mlacp load-balance secondary vlan 30,100
Router(config-if)# end
```

Standby PoA-POA2

```
Router# configure terminal
Router(config)# interface port-channel1
Router(config-if)# mlacp interchassis group 1
Router(config-if)# mlacp mode active-active
Router(config-if)# mlacp load-balance primary vlan 30,100
Router(config-if)# mlacp load-balance secondary vlan 10,20
Router(config-if)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Carrier Ethernet commands	Cisco IOS Carrier Ethernet Command Reference

Standards and RFCs

Standard/RFC	Title
IEEE 802.3ad	*Link Aggregation Control Protocol*
IEEE 802.1ak	*Multiple Registration Protocol*
RFC 4447	*Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*
RFC 4762	*Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*

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

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 ICCP Multichassis VLAN Redundancy

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 27: Feature Information for ICCP Multiichassis VLAN Redundancy

Feature Name	Releases	Feature Information
ICCP Multichassis VLAN Redundancy	15.1(3)S	Pseudo mLACP provides a flexible dual-homing redundancy mechanism. It uses similar principles as mLACP, but without the implementation of LACP between the PEs and CEs. The PE ports are configured in such a way that they act as if connected to a virtual device over an MC-LAG with mLACP. Ports can be placed in active/active mode with manual VLAN load balancing. The following commands were introduced or modified: **debug lacp**, **debug mvrp**, **mlacp load-balance**, **mlacp load-balance force-switchover**, **mlacp mode active-active**, **mlacp reversion-delay**, **show lacp**.

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.

CHAPTER 20

ITU-T G.8032 Ethernet Ring Protection Switching

The ITU-T G.8032 Ethernet Ring Protection Switching feature implements protection switching mechanisms for Ethernet layer ring topologies. This feature uses the G.8032 Ethernet Ring Protection (ERP) protocol, defined in ITU-T G.8032, to provide protection for Ethernet traffic in a ring topology, while ensuring that no loops are within the ring at the Ethernet layer. The loops are prevented by blocking traffic on either a predetermined link or a failed link.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 G.8032 Ethernet Ring Protection Switching

- The Ethernet Flow Points (EFPs) must be configured.

About ITU-T G.8032 Ethernet Ring Protection Switching

Ring Protection Links

An Ethernet ring consists of multiple Ethernet ring nodes. Each Ethernet ring node is connected to adjacent Ethernet ring nodes using two independent ring links. A ring link prohibits formation of loops that affect the network. The Ethernet ring uses a specific link to protect the entire Ethernet ring. This specific link is called the Ring Protection Link (RPL). A ring link is bound by two adjacent Ethernet ring nodes and a port for a ring link (also known as a ring port). There must be at least two Ethernet ring nodes in a Ethernet ring.

ITU-T G.8032 Ethernet Ring Protection Switching Functionality

The Ethernet ring protection functionality includes the following:

- Loop avoidance
- The use of learning, forwarding, and Filtering Database (FDB) mechanisms

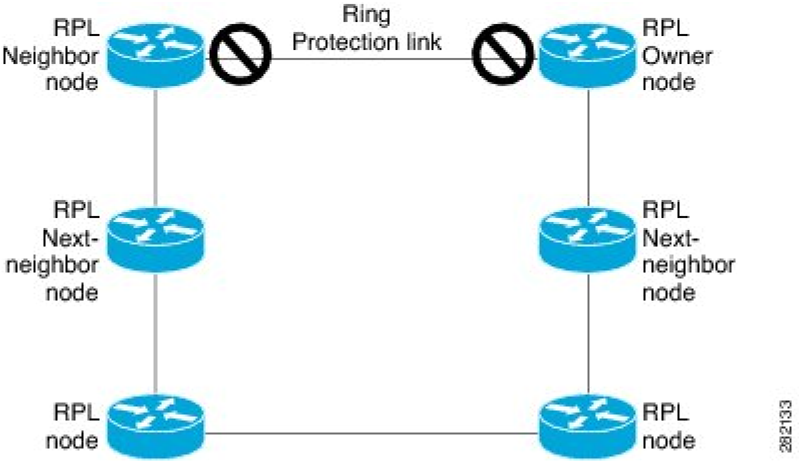
Loop avoidance in an Ethernet ring is achieved by ensuring that, at any time, traffic flows on all but the Ring Protection Link (RPL).

The following is a list of RPL types (or RPL nodes) and their functions:

- RPL owner—Responsible for blocking traffic over the RPL so that no loops are formed in the Ethernet traffic. There can be only one RPL owner in a ring.
- RPL neighbor node—An Ethernet ring node adjacent to the RPL. It is responsible for blocking its end of the RPL under normal conditions. This node type is optional and prevents RPL usage when protected.
- RPL next-neighbor node—Next-neighbor node is an Ethernet ring node adjacent to an RPL owner node or RPL neighbor node. It is mainly used for FDB flush optimization on the ring. This node is also optional.

The following figure illustrates the G.8032 Ethernet ring topology.

Figure 8: G.8032 Ethernet Ring Topology



R-APS Control Messages

Nodes on the ring use control messages called Ring Automatic Protection Switching (R-APS) messages to coordinate the activities of switching the ring protection link (RPL) on and off. Any failure along the ring triggers a R-APS Signal Failure (R-APS SF) message in both directions of the nodes adjacent to the failed link, after the nodes have blocked the port facing the failed link. On obtaining this message, the RPL owner unblocks the RPL port.

Note A single link failure in the ring ensures a loop-free topology.

CFM Protocols and Link Failures

Connectivity Fault Management (CFM) and line status messages are used to detect ring link and node failure. During the recovery phase, when the failed link is restored, the nodes adjacent to the restored link send Ring Automatic Protection Switching (R-APS) No Request (R-APS NR) messages. On obtaining this message, the ring protection link (RPL) owner blocks the RPL port and sends R-APS NR and R-APS RPL (R-APS NR, RB) messages. These messages cause all other nodes, other than the RPL owner in the ring, to unblock all blocked ports. The Ethernet Ring Protection (ERP) protocol works for both unidirectional failure and multiple link failure scenarios in a ring topology.

Note The G.8032 Ethernet Ring Protection (ERP) protocol uses CFM Continuity Check Messages (CCMs) at an interval of 3.3 milliseconds (ms). At this interval (which is supported only on selected platforms), SONET-like switching time performance and loop-free traffic can be achieved.

G.8032 Ring-Supported Commands and Functionality

A G.8032 ring supports these basic operator administrative commands:

- Force switch (FS)—Allows the operator to forcefully block a particular ring port. Note the following points about FS commands:
    - Effective even if there is an existing SF condition
    - Multiple FS commands for ring are supported
    - May be used to allow immediate maintenance operations
- Manual switch (MS)—Allows the operator to manually block a particular ring port. Note the following points about MS commands:
    - Ineffective in an existing FS or signal failure (SF) condition
    - Overridden by new FS or SF conditions
    - Multiple MS commands cancel all MS commands
- Clear—Cancels an existing FS or MS command on the ring port. The Clear command is used at the ring protection link (RPL) owner to clear a nonrevertive mode condition.

A G.8032 ring can support multiple instances. An instance is a logical ring running over a physical ring. Such instances are used for various reasons, such as load-balancing VLANs over a ring. For example, odd-numbered VLANs may go in one direction of the ring, and even-numbered VLANs may go in the other direction. Specific VLANs can be configured under only one instance. They cannot overlap multiple instances. Otherwise, data traffic or Ring Automatic Protection Switching (R-APS) messages may cross logical rings, which is not desirable.

G.8032 ERP Timers

The G.8032 Ethernet Ring Protection (ERP) protocol specifies the use of different timers to avoid race conditions and unnecessary switching operations:

- Delay timers—Used by the Ring Protection Link (RPL) owner to verify that the network has stabilized before blocking the RPL. Note the following points about delay timers.
    - After a signal failure (SF) condition, a Wait-to-Restore (WTR) timer is used to verify that the SF is not intermittent.
    - The WTR timer can be configured by the operator. The default time interval is 5 minutes; the time interval ranges from 1 to 12 minutes.
    - After a force switch (FS) or a manual switch (MS) command is issued, a Wait-to-Block (WTB) timer is used to verify that no background condition exists.

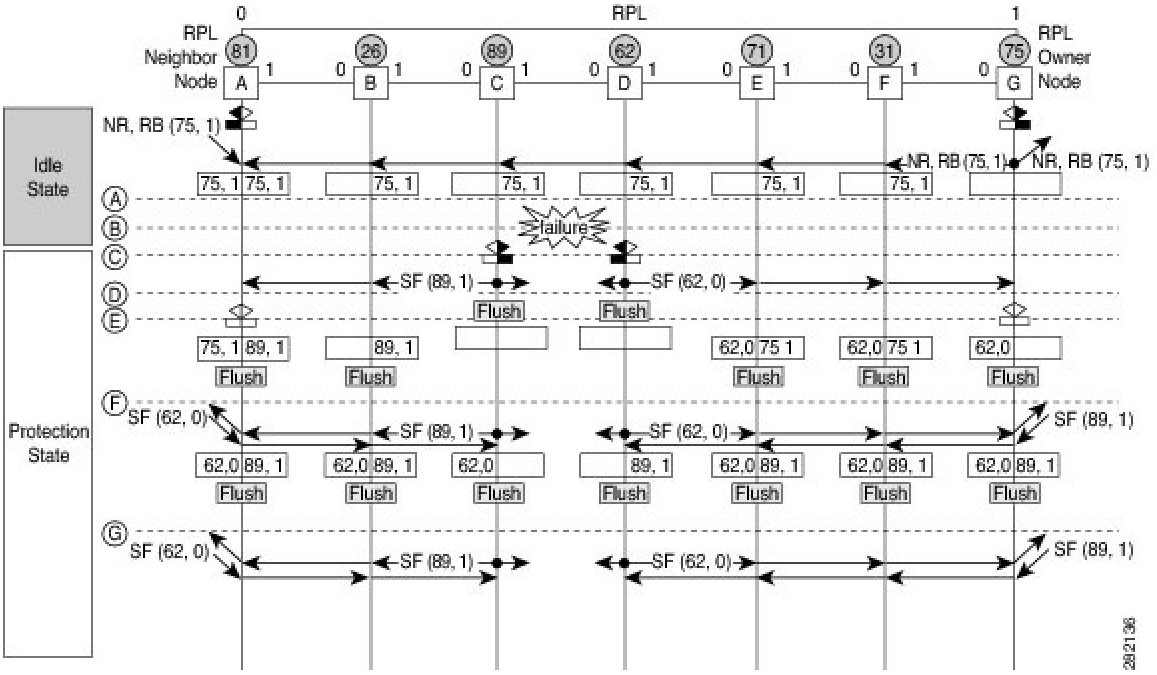
Note The WTB timer interval may be shorter than the WTR timer interval.

- Guard timer—Used by all nodes when changing state; the guard timer blocks latent outdated messages from causing unnecessary state changes. The guard timer can be configured. The default time interval is 500 ms; the time interval ranges from 10 to 2000 ms.
- Hold-off timers—Used by the underlying Ethernet layer to filter out intermittent link faults. The hold-off timer can be configured. The default time interval is 0 seconds; the time interval ranges from 0 to 10 seconds. Faults are reported to the ring protection mechanism only if this timer expires.

Protection Switching Functionality in a Single Link Failure and Recovery

The following figure illustrates protection switching functionality in a single-link failure.

Figure 9: G.8032 Ethernet Ring Protection Switching in a Single-Link Failure



The figure represents an Ethernet ring topology consisting of seven Ethernet ring nodes. The ring protection link (RPL) is the ring link between Ethernet ring nodes A and G. In this topology, both ends of the RPL are blocked. Ethernet ring node G is the RPL owner node, and Ethernet ring node A is the RPL neighbor node.

The following sequence describes the steps followed in the single-link failure:

1. A link operates in the normal condition.
2. A failure occurs.

3 Ethernet ring nodes C and D detect a local signal failure (SF) condition and after the hold-off time interval, block the failed ring port and perform the FDB flush.

4 Ethernet ring nodes C and D start sending Ring Automatic Protection Switching (R-APS) SF messages periodically along with the (node ID and bidirectional path-protected ring (BPR) identifier pair) on both ring ports while the SF condition persists.

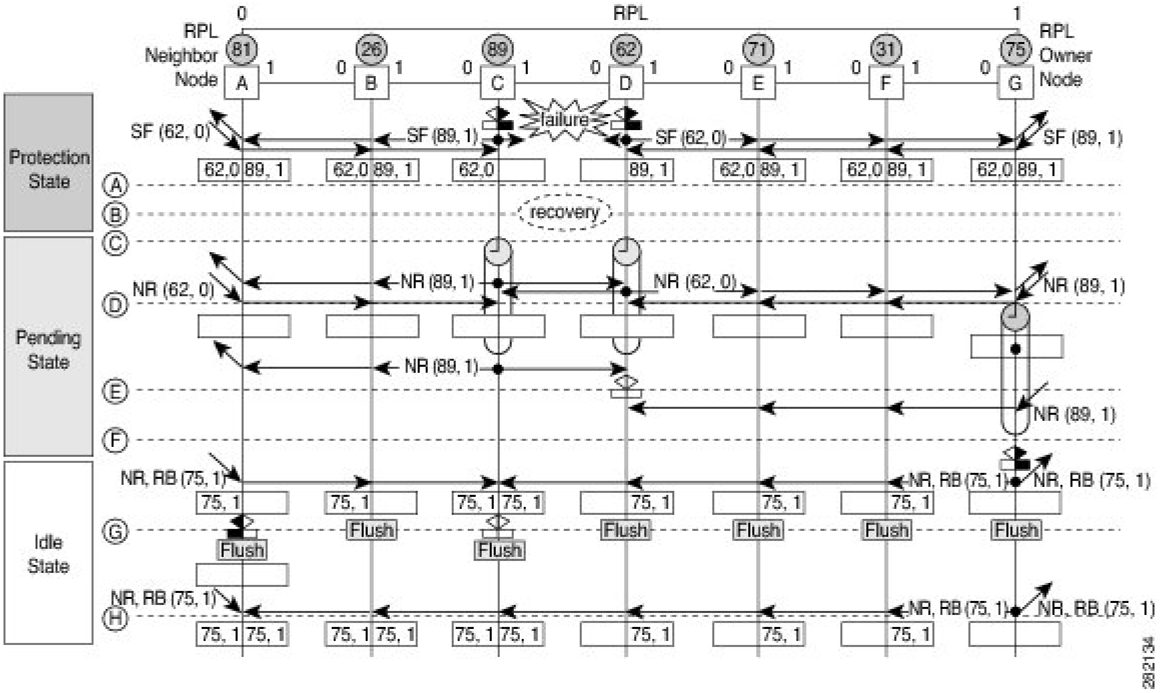
5 All Ethernet ring nodes receiving an R-APS SF message perform the FDB flush. When the RPL owner node G and RPL neighbor node A receive an R-APS SF message, the Ethernet ring node unblocks its end of the RPL and performs the FDB flush.

6 All Ethernet ring nodes receiving a second R-APS SF message perform the FDB flush again; the additional FDB flush is because of the node ID and BPR-based configuration.

7 R-APS SF messages are detected on the Ethernet Ring indicating a stable SF condition. Further R-APS SF messages trigger no further action.

The following figure illustrates the steps taken in a revertive operation in a single-link failure.

Figure 10: Single-Link Failure Recovery (Revertive Operation)



The following sequence describes the steps followed in the single-link failure revertive (recovery) operation:

1 A link operates in the stable SF condition.

2 Recovery of link failure occurs.

3 Ethernet ring nodes C and D detect clearing of the SF condition, start the guard timer, and initiate periodic transmission of the R-APS No Request (NR) messages on both ring ports. (The guard timer prevents the reception of R-APS messages.)

4 When the Ethernet ring nodes receive an R-APS NR message, the node ID and BPR identifier pair of a receiving ring port is deleted and the RPL owner node starts the Wait-to-Restore (WTR) timer.

5 When the guard timer expires on Ethernet ring nodes C and D, the nodes may accept the new R-APS messages, if any. Ethernet ring node D receives an R-APS NR message with a higher node ID from Ethernet ring node C, and unblocks its nonfailed ring port.

6 When the WTR timer expires, the RPL owner node blocks its end of the RPL, sends R-APS (NR or route blocked [RB]) message with the (node ID and BPR identifier pair), and performs the FDB flush.

7 When Ethernet ring node C receives an R-APS (NR or RB) message, the node removes the block on its blocked ring ports, and stops sending R-APS NR messages. On the other hand, when the RPL neighbor node A receives an R-APS NR or RB message, the node blocks its end of the RPL. In addition, Ethernet ring nodes A to F perform the FDB flush when receiving an RAPS NR or RB message because of the node ID and BPR-based configuration.

Ethernet Flow Points

An Ethernet flow point (EFP) is a forwarding decision point in the provider edge (PE) router, which gives network designers flexibility to make many Layer 2 flow decisions within the interface. Many EFPs can be configured on a single physical port. (The number varies from one device to another.) EFPs are the logical demarcation points of an Ethernet virtual connection (EVC) on an interface. An EVC that uses two or more user network interfaces (UNIs) requires an EFP on the associated ingress and egress interfaces of every device that the EVC passes through.

EFPs can be configured on any Layer 2 traffic port; however, they are usually configured on UNI ports. The following parameters (matching criteria) can be configured on the EFP:

- Frames of a specific VLAN, a VLAN range, or a list of VLANs (100-150 or 100,103,110)
- Frames with no tags (untagged)
- Frames with identical double-tags (VLAN tags) as specified
- Frames with identical Class of Service (CoS) values

A frame passes each configured match criterion until the correct matching point is found. If a frame does not fit any of the matching criteria, it is dropped. Default criteria can be configured to avoid dropping frames.

The following types of commands can be used in an EFP:

- Rewrite commands—In each EFP, VLAN tag management can be specified with the following actions:
    - Pop—1) pops out a tag; 2) pops out two tags
    - Push—1) pushes in a tag; 2) pushes in two tags
    - Translate—1 to 1) changes a tag value; 1 to 2) pops one tag and pushes two tags; 2 to 1) pops two tags and pushes one tag; 2 to 2) changes the value for two tags

- Forwarding commands—Each EFP specifies the forwarding command for the frames that enter the EFP. Only one forwarding command can be configured per EFP. The forwarding options are as follows:
    - Layer 2 point-to-point forwarding to a pseudowire tunnel
    - Multipoint bridge forwarding to a bridge domain entity

  - Local switch-to-switch forwarding between two different interfaces

- Feature commands—In each EFP, the QoS features or parameters can be changed and the ACL can be updated.

Service Instances and Associated EFPs

Configuring a service instance on a Layer 2 port creates a pseudoport or EFP on which you configure EVC features. Each service instance has a unique number per interface, but you can use the same number on different interfaces because service instances on different ports are not related.

An EFP classifies frames from the same physical port to one of the multiple service instances associated with that port, based on user-defined criteria. Each EFP can be associated with different forwarding actions and behavior.

When an EFP is created, the initial state is UP. The state changes to DOWN under the following circumstances:

- The EFP is explicitly shut down by a user.
- The main interface to which the EFP is associated is down or removed.
- If the EFP belongs to a bridge domain, the bridge domain is down.
- The EFP is forced down as an error-prevention measure of certain features.

Use the **service instance ethernet** interface configuration command to create an EFP on a Layer 2 interface and to enter service instance configuration mode. Service instance configuration mode is used to configure all management and control data plane attributes and parameters that apply to the service instance on a per-interface basis. The service instance number is the EFP identifier.

After the device enters service instance configuration mode, you can configure these options:

- default--Sets a command to its defaults
- description--Adds a service instance-specific description
- encapsulation--Configures Ethernet frame match criteria
- exit--Exits from service instance configuration mode
- no--Negates a command or sets its defaults
- shutdown--Takes the service instance out of service

How to Configure ITU-T G.8032 Ethernet Ring Protection Switching

Configuring the Ethernet Ring Profile

To configure the Ethernet ring profile, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet ring g8032 profile** *profile-name*
4. **timer** {**guard** *seconds* | **hold-off** *seconds* | **wtr** *minutes*}
5. **non-revertive**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet ring g8032 profile** *profile-name* **Example:** `Device(config)# ethernet ring g8032 profile profile1`	Creates the Ethernet ring profile and enters Ethernet ring profile configuration mode.

Step 4	**timer** {**guard** *seconds* | **hold-off** *seconds* | **wtr** *minutes*}
	Example:
	`Device(config-erp-profile)# timer hold-off 5`
		Specifies the time interval for the guard, hold-off, and Wait-to-Restore (WTR) timers.

Step 5	**non-revertive** **Example:** `Device(config-erp-profile)# non-revertive`	Specifies a nonrevertive Ethernet ring instance. • By default, Ethernet ring instances are revertive.

Step 6	**end**
	Example:
	`Device(config-erp-profile)# end`
		Returns to user EXEC mode.

Configuring Ethernet CFM MEPs

Configuring Ethernet Connectivity Fault Management (CFM) maintenance endpoints (MEPs) is optional although recommended for fast failure detection and CFM monitoring. When CFM monitoring is configured, note the following points:

- Static remote MEP (RMEP) checking should be enabled.
- The MEPs should be configured to enable Ethernet fault detection.

For information about configuring Ethernet Connectivity Fault Management (CFM) maintenance endpoints (MEPs), see the "Configuring Ethernet Connectivity Fault Management in a Service Provider Network" module of the *Carrier Ethernet Configuration Guide*.

Enabling Ethernet Fault Detection for a Service

To enable Ethernet Fault Detection (EFD) for a service to achieve fast convergence, complete the following steps

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm global**
4. **link-protection enable**
5. **link-protection group management vlan** *vlan-id*
6. **link-protection group** *group-number* **pccm vlan** *vlan-id*
7. **ethernet cfm domain***domain-name* **level** *level-id* [**direction outward**]
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. **efd notify g8032**
11. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables Ethernet CFM globally.
Step 4	**link-protection enable** **Example:** `Device(config)# link-protection enable`	Enables link protection globally on the router.
Step 5	**link-protection group management vlan** *vlan-id* **Example:** `Device(config)# link-protection group management vlan 51`	Defines the management VLAN used for link protection.
Step 6	**link-protection group** *group-number* **pccm vlan** *vlan-id* **Example:** `Device(config)# link-protection group 2 pccm vlan 16`	Specifies an ODU-to-ODU continuity check message (P-CCM) VLAN.
Step 7	**ethernet cfm domain***domain-name* **level** *level-id* [**direction outward**] **Example:** `Device(config)# ethernet cfm domain G8032 level 4`	Configures the CFM domain for ODU 1 and enters Ethernet CFM configuration mode.
Step 8	**service** {*ma-name* | *ma-num* | **vlan-id** *vlan-id* | **vpn-id** *vpn-id*} [**port** | **vlan** *vlan-id* [**direction down**]] **Example:** `Device(config-ecfm)# service 8032_service evc 8032-evc vlan 1001 direction down`	Defines a maintenance association for ODU 1 and enters Ethernet CFM service instance configuration mode.

	Command or Action	Purpose
Step 9	**continuity-check** [**interval** *time* | **loss-threshold** *threshold* | **static rmep**] **Example:** `Device(config-ecfm-srv)# continuity-check interval 3.3ms`	Enables the transmission of continuity check messages (CCMs).
Step 10	**efd notify g8032** **Example:** `Device(config-ecfm-srv)# efd notify g8032`	Enables CFM to notify registered protocols when a defect is detected or cleared, which matches the current fault alarm priority.

Step 11	**end**
	Example:
	`Device(config-ecfm-srv)# end`
		Returns to user EXEC mode.

Configuring the Ethernet Protection Ring

To configure the Ethernet Protection Ring (EPR), complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet ring g8032** *ring-name*
4. **port0 interface** *type number*
5. **monitor service instance** *instance-id*
6. **exit**
7. **port1** {**interface***type number* | **none**}
8. **monitor service instance** *instance-id*
9. **exit**
10. **exclusion-list vlan-ids** *vlan-id*
11. **open-ring**
12. **instance** *instance-id*
13. **description** *descriptive-name*
14. **profile** *profile-name*
15. **rpl** {**port0** | **port1**} {**owner** | **neighbor** | **next-neighbor** }
16. **inclusion-list vlan-ids** *vlan-id*
17. **aps-channel**
18. **level** *level-value*
19. **port0 service instance** *instance-id*
20. **port1 service instance** {*instance-id* | **none** }
21. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet ring g8032** *ring-name* **Example:** `Device(config)# ethernet ring g8032 ring1`	Specifies the Ethernet ring and enters Ethernet ring port configuration mode.

	Command or Action	Purpose
Step 4	**port0 interface** *type number* **Example:** `Device(config-erp-ring)# port0 interface fastethernet 0/1/0`	Connects port0 of the local node of the interface to the Ethernet ring and enters Ethernet ring protection mode.
Step 5	**monitor service instance** *instance-id* **Example:** `Device(config-erp-ring-port)# monitor service instance 1`	Assigns the Ethernet service instance to monitor the ring port (port0) and detect ring failures.
Step 6	**exit** **Example:** `Device(config-erp-ring-port)# exit`	Exits Ethernet ring port configuration mode.

Step 7	**port1** {**interface***type number* | **none**}
	Example:
	`Device(config-erp-ring)# port1 interface fastethernet 0/1/1`
		Connects port1 of the local node of the interface to the Ethernet ring and enters Ethernet ring protection mode.

Step 8	**monitor service instance** *instance-id* **Example:** `Device(config-erp-ring-port)# monitor service instance 2`	Assigns the Ethernet service instance to monitor the ring port (port1) and detect ring failures. • The interface (to which port1 is attached) must be a subinterface of the main interface.
Step 9	**exit** **Example:** `Device(config-erp-ring-port)# exit`	Exits Ethernet ring port configuration mode.
Step 10	**exclusion-list vlan-ids** *vlan-id* **Example:** `Device(config-erp-ring)# exclusion-list vlan-ids 2`	Specifies VLANs that are unprotected by the Ethernet ring protection mechanism.
Step 11	**open-ring** **Example:** `Device(config-erp-ring)# open-ring`	Specifies the Ethernet ring as an open ring.

	Command or Action	Purpose
Step 12	**instance** *instance-id* **Example:** `Device(config-erp-ring)# instance 1`	Configures the Ethernet ring instance and enters Ethernet ring instance configuration mode.
Step 13	**description** *descriptive-name* **Example:** `Device(config-erp-inst)# description cisco_customer_instance`	Specifies a descriptive name for the Ethernet ring instance.
Step 14	**profile** *profile-name* **Example:** `Device(config-erp-inst)# profile profile1`	Specifies the profile associated with the Ethernet ring instance.

Step 15	**rpl** {**port0** | **port1**} {**owner** | **neighbor** | **next-neighbor** }
	Example:
	`Device(config-erp-inst)# rpl port0 neighbor`
		Specifies the Ethernet ring port on the local node as the RPL owner, neighbor, or next neighbor.

Step 16	**inclusion-list vlan-ids** *vlan-id* **Example:** `Device(config-erp-inst)# inclusion-list vlan-ids 11`	Specifies VLANs that are protected by the Ethernet ring protection mechanism.
Step 17	**aps-channel** **Example:** `Device(config-erp-inst)# aps-channel`	Enters Ethernet ring instance aps-channel configuration mode.
Step 18	**level** *level-value* **Example:** `Device(config-erp-inst-aps)# level 5`	Specifies the Automatic Protection Switching (APS) message level for the node on the Ethernet ring. • All nodes in the Ethernet ring must be configured with the same level.
Step 19	**port0** **service instance** *instance-id* **Example:** `Device(config-erp-inst-aps)# port0 service instance 100`	Associates APS channel information with port0.

	Command or Action	Purpose

Step 20	**port1 service instance** {*instance-id* | **none** }
	Example:
	`Device(config-erp-inst-aps)# port1 service instance 100`
		Associates APS channel information with port1.

Step 21	**end**
	Example:
	`Device(config-erp-inst-aps)# end`
		Returns to user EXEC mode.

Configuring Topology Change Notification Propagation

To configure topology change notification (TCN) propagation, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet tcn-propagation G8032 to {REP | G8032}**
4. **end**

DETAILED STEPS

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

Step 3	**ethernet tcn-propagation G8032 to {REP | G8032}**
	Example:
	`Device(config)# ethernet tcn-propagation G8032 to G8032`
		Allows topology change notification (TCN) propagation from a source protocol to a destination protocol.
		• Source and destination protocols vary by platform and release.

	Command or Action	Purpose

Step 4	**end**
	Example:
	`Device(config)# end`
		Returns to user EXEC mode.

Configuring a Service Instance

To configure a service instance, complete the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service instance** *instance-id* **ethernet** [*evc-id*]
5. **encapsulation dot1q** *vlan-id* [**native**]
6. **bridge-domain** *bridge-id* [**split-horizon** [**group** *group-id*]]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 4/0/0`	Specifies the interface type and number.

	Command or Action	Purpose
Step 4	**service instance** *instance-id* **ethernet** [*evc-id*] **Example:** `Device(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:** `Device(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:** `Device(config-if-srv)# bridge-domain 12`	Binds the service instance to a bridge domain instance.

Step 7	**end**
	Example:
	`Device(config-if-srv)# end`
		Exits service instance configuration mode.

Verifying the Ethernet Ring Protection (ERP) Switching Configuration

To verify the ERP switching configuration, use one or more of the following commands in any order.

SUMMARY STEPS

1. **enable**
2. **show ethernet ring g8032 status** [*ring-name*] [**instance** [*instance-id*]]
3. **show ethernet ring g8032 brief** [*ring-name*] [**instance** [*instance-id*]]
4. **show ethernet ring g8032 summary**
5. **show ethernet ring g8032 statistics** [*ring-name*] [**instance** [*instance-id*]]
6. **show ethernet ring g8032 profile** [*profile-name*]
7. **show ethernet ring g8032 port status interface** [*type number*]
8. **show ethernet ring g8032 configuration** [*ring-name*] **instance** [*instance-id*]
9. **show ethernet ring g8032 trace** {**ctrl** [*ring-name* **instance** *instance-id*] | **sm**}
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ethernet ring g8032 status** [*ring-name*] [**instance** [*instance-id*]] **Example:** `Device# show ethernet ring g8032 status RingA instance 1`	Displays a status summary for the ERP instance.
Step 3	**show ethernet ring g8032 brief** [*ring-name*] [**instance** [*instance-id*]] **Example:** `Device# show ethernet ring g8032 brief`	Displays a brief description of the functional state of the ERP instance.
Step 4	**show ethernet ring g8032 summary** **Example:** `Device# show ethernet ring g8032 summary`	Displays a summary of the number of ERP instances in each state of the ERP switching process.
Step 5	**show ethernet ring g8032 statistics** [*ring-name*] [**instance** [*instance-id*]] **Example:** `Device# show ethernet ring g8032 statistics RingA instance 1`	Displays the number of events and Ring Automatic Protection Switching (R-APS) messages received for an ERP instance.
Step 6	**show ethernet ring g8032 profile** [*profile-name*] **Example:** `Device# show ethernet ring g8032 profile gold`	Displays the settings for one or more ERP profiles.
Step 7	**show ethernet ring g8032 port status interface** [*type number*] **Example:** `Device# show ethernet ring g8032 port status interface fastethernet 0/0/1`	Displays Ethernet ring port status information for the interface.

	Command or Action	Purpose
Step 8	**show ethernet ring g8032 configuration** [*ring-name*] **instance** [*instance-id*] **Example:** `Device# show ethernet ring g8032 configuration RingA instance 1`	Displays the details of the ERP instance configuration manager.
Step 9	**show ethernet ring g8032 trace** {**ctrl** [*ring-name* **instance** *instance-id*] | **sm**} **Example:** `Device# show ethernet ring g8032 trace sm`	Displays information about ERP traces.

Step 10	**end**
	Example:
	`Device# end`
		Returns to privileged EXEC mode.

Configuration Examples for ITU-T G.8032 Ethernet Ring Protection Switching

Example: Configuring Ethernet Ring Protection Switching

The following is an example of an Ethernet Ring Protection (ERP) switching configuration:

```
ethernet ring g8032 profile profile_ABC
 timer wtr 1
 timer guard 100
 timer hold-off  1

ethernet ring g8032 major_ring_ABC
 exclusion-list vlan-ids 1000
 port0 interface FastEthernet 0/0/0
  monitor service instance 103
 port1 interface FastEthernet 0/1/0
  monitor service instance 102
 instance 1
  profile profile_ABC
  rpl port0 owner
  inclusion-list vlan-ids 100
  aps-channel
   port0 service instance 100
   port1 service instance 100
  !
interface FastEthernet 0/0/0
 no ip address
 service instance 100 ethernet
```

```
  encapsulation dot1q 100
  bridge-domain 100
service instance 200 ethernet
  encapsulation dot1q 200
  bridge-domain 200

 !
!
interface FastEthernet 0/1/1
 no ip address
 service instance 100 ethernet
  encapsulation dot1q 100
  bridge-domain 100
service instance 200 ethernet
  encapsulation dot1q 200
  bridge-domain 200

 !
!
```

Example: Enabling Ethernet Fault Detection for a Service

```
ethernet cfm domain G8032 level 4
service 8032_service evc 8032-evc vlan 1001 direction down
  continuity-check
  continuity-check interval 3.3ms
  offload sampling 1000
  efd notify g8032
ethernet ring g8032 profile TEST
timer wtr 1
timer guard 100
ethernet ring g8032 open
open-ring
port0 interface GigabitEthernet0/1/3
  monitor service instance 1001
port1 none
instance 1
  profile TEST
  inclusion-list vlan-ids 2-500,1001
  aps-channel
   port0 service instance 1001
   port1 none
  !
!
instance 2
  profile TEST
  rpl port0 owner
  inclusion-list vlan-ids 1002,1005-2005
  aps-channel
   port0 service instance 1002
   port1 none
  !

interface GigabitEthernet0/1/3
no ip address
load-interval 30
shutdown
negotiation auto
storm-control broadcast level 10.00
storm-control multicast level 10.00
storm-control unicast level 90.00
service instance 1 ethernet
  encapsulation untagged
  l2protocol peer lldp
  bridge-domain 1
!
service instance trunk 10 ethernet
  encapsulation dot1q 2-500,1005-2005
  rewrite ingress tag pop 1 symmetric
  bridge-domain from-encapsulation
!
service instance 1001 ethernet 8032-evc
  encapsulation dot1q 1001
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1001
  cfm mep domain G8032 mpid 20
!
service instance 1002 ethernet 8032-evc-1
  encapsulation dot1q 1002
  rewrite ingress tag pop 1 symmetric
  bridge-domain 1002
!
End
```

Example: Verifying the Ethernet Ring Protection Configuration

The following is sample output from the **show ethernet ring g8032 configuration** command. Use this command to verify if the configuration entered is valid and to check for any missing configuration parameters.

```
Device# show ethernet ring g8032 configuration

ethernet ring ring0
 Port0: GigabitEthernet0/0/0 (Monitor: GigabitEthernet0/0/0)
 Port1: GigabitEthernet0/0/4 (Monitor: GigabitEthernet0/0/4)
 Exclusion-list VLAN IDs: 4001-4050
 Open-ring: no
 Instance 1
  Description:
  Profile:     opp
  RPL:
  Inclusion-list VLAN IDs: 2,10-500
  APS channel
   Level: 7
   Port0: Service Instance 1
   Port1: Service Instance 1
  State: configuration resolved
```

Additional References for ITU-T G.8032 Ethernet Ring Protection Switching

Related Documents

Related Topic	Document Title
Ethernet Connectivity Fault Management (CFM)	*Configuring Ethernet Connectivity Fault Management in a Service Provider Network*
G.8032 Ethernet Ring Protection (ERP) administrative procedures	http://docwiki.cisco.com/wiki/G.8032_Ethernet_Ring_Protection_(ERP)_Administrative_Procedures
Carrier Ethernet commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Carrier Ethernet Command Reference*

Related Topic	Document Title
Cisco IOS commands: master list of commands with complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS Master Commands List, All Releases

Standards

Standard	Title
ITU-T	*ITU-T Y.1731 OAM Mechanisms for Ethernet-Based Networks*

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 ITU-T G.8032 Ethernet Ring Protection Switching

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 28: Feature Information for ITU-T G.8032 Ethernet Ring Protection Switching

Feature Name	Releases	Feature Information
ITU-T G.8032 Ethernet Ring Protection Switching	15.2(4)S 15.3(1)S	

Feature Name	Releases	Feature Information
		The ITU-T G.8032 Ethernet Ring Protection Switching feature implements protection switching mechanisms for Ethernet layer ring topologies. This feature uses the G.8032 Ethernet Ring Protection (ERP) protocol, defined in ITU-T G.8032, to provide protection for Ethernet traffic in a ring topology, while ensuring that no loops are within the ring at the Ethernet layer. The loops are prevented by blocking traffic on either a predetermined link or a failed link. The following commands were introduced or modified: **aps-channel**, **clear ethernet ring g8032 statistics**, **debug ethernet ring g8032 errors**, **debug ethernet ring g8032 events**, **debug ethernet ring g8032 fsm**, **debug ethernet ring g8032 packets**, **description (Ethernet ring)**, **ethernet ring g8032**, **ethernet ring g8032 profile**, **ethernet tcn-propagation**, **exclusion-list**, **inclusion-list**, **instance (Ethernet ring)**, **level**, **monitor service instancenon-revertiveopen-ring**, **port0**, **port0 service instance**, **port1**, **port1 service instance**, **profile**, **rpl**, **show ethernet cfm domain**, **show ethernet cfm errors**,**show ethernet cfm maintenance-points remote**, **show ethernet cfm maintenance-points remote crosscheck**, **show ethernet ring g8032 brief**, **show ethernet ring g8032 configuration**, **show ethernet ring g8032 port status**, **show ethernet ring g8032 profile**, **show ethernet ring g8032 statistics**, **show ethernet ring g8032 status**, **show ethernet ring g8032 summary** , **show ethernet ring g8032 trace**,**show ethernet service instance** and **timer (Ethernet ring)**.

CHAPTER 21

Configuring IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations

This module describes how to configure an IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) operation to gather the following performance measurements for Ethernet service:

- Ethernet Delay
- Ethernet Delay Variation
- Ethernet Frame Loss Ratio

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 ITU-T Y.1731 Operations

IEEE-compliant Connectivity Fault Management (CFM) must be configured and enabled for Y.1731 performance monitoring to function.

Restrictions for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731)

- Depending on your Cisco software release, SNMP is not supported for reporting threshold events or collecting performance statistics for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) operations.
- Continuity Check Message (CCM)-based dual-ended Ethernet frame loss operations are not supported.
- In a single-ended Ethernet operation, performance measurement statistics can be retrieved only at the device on which the sender Ethernet Connectivity Fault Management (CFM) Maintenance End Point (MEP) is configured.
- Frame Loss Measurement is not supported on Cisco ME 3600X Series and 3800X Series Ethernet Access Switches.
- P2 IMs are to be used for CFM and Y1731
- Do not configure rewrite on the EFPs throughout the l2 circuit to avoid losing the cos value.
- CFMoXconnect on ASR903 works only if the control-word is switched on. To start DM timestamping, switch on the control-word if the remote end is not switched on.
- To avoid errors in RX and TX timestamping, ensure to have Y1731 sender as PTP master, and the Y1731 responder as PTP slave.
- Reconfigure IP SLA Y1731 while doing online insertion removal (OIR) of IM or router reload because local MEP is deleted during the course.

Configuring IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations

This module describes how to configure an IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) operation to gather the following performance measurements for Ethernet service:

- Ethernet Delay
- Ethernet Delay Variation
- Ethernet Frame Loss Ratio

How to Configure IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations

Configuring a Dual-Ended Ethernet Delay or Delay Variation Operation

Perform the tasks for configuring a dual-ended operation in the order presented.

Note To remove the MEP configurations in an already-configured dual-ended operation, always remove the MEPs in the reverse order in which they were configured. That is, remove the scheduler first, then the threshold monitoring configuration, and then the sender MEP configuration on the source device before removing the scheduler, proactive threshold monitoring, and receiver MEP configuration on the destination device.

Configuring a Receiver MEP on the Destination Device

Before You Begin

Time synchronization is required between the source and destination devices in order to provide accurate one-way delay (latency) or delay-variation measurements. Configure either Precision Time Protocol (PTP) or Network Time Protocol (NTP) on both the source and destination devices.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **ethernet y1731 delay receive 1DM domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} **cos** *cos* {**mpid** *source-mp-id* | **mac-address** *source-address*}
5. **aggregate interval** *seconds*
6. **distribution** {**delay** | **delay-variation**} **one-way** *number-of-bins boundary*[**,**...**,***boundary*]
7. **frame offset** *offset-value*
8. **history interval** *intervals-stored*
9. **max-delay** *milliseconds*
10. **owner** *owner-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.

	Command or Action	Purpose
Step 3	**ip sla** *operation-number* **Example:** `Router(config-term)# ip sla 501`	Begins configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4	**ethernet y1731 delay receive 1DM domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} **cos** *cos* {**mpid** *source-mp-id* | **mac-address** *source-address*}
	Example:
	`Router(config-ip-sla)# ethernet y1731 delay receive 1DM domain xxx evc yyy cos 3 mpid 101`
		Begins configuring the receiver on the responder and enters IP SLA Y.1731 delay configuration mode.
		• The *source-mp-id* or *source-address* configured by this command corresponds to that of the MEP being configured.

Step 5	**aggregate interval** *seconds* **Example:** `Router(config-sla-y1731-delay)# aggregate interval 900`	(Optional) Configures the length of time during which the performance measurements are conducted and the results stored.
Step 6	**distribution** {**delay** | **delay-variation**} **one-way** *number-of-bins boundary*[**,**...**,***boundary*] **Example:** `Router(config-sla-y1731-delay)# distribution delay-variation one-way 5 5000,10000,15000,20000,-1`	(Optional) Specifies measurement type and configures bins for statistics distributions kept.
Step 7	**frame offset** *offset-value* **Example:** `Router(config-sla-y1731-delay)# frame offset 1`	(Optional) Sets the value for calculating delay variation rates.
Step 8	**history interval** *intervals-stored* **Example:** `Router(config-sla-y1731-delay)# history interval 2`	(Optional) Sets the number of statistics distributions kept during the lifetime of an IP SLAs Ethernet operation.

	Command or Action	Purpose
Step 9	**max-delay** *milliseconds* **Example:** `Router(config-sla-y1731-delay)# max-delay 5000`	(Optional) Sets the amount of time an MEP waits for a frame.
Step 10	**owner** *owner-id* **Example:** `Router(config-sla-y1731-delay)# owner admin`	(Optional) Configures the owner of an IP SLAs operation.

Step 11	**end**
	Example:
	`Router(config-sla-y1731-delay)# end`
		Exits to privileged EXEC mode.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps, see the "Configuring Proactive Threshold Monitoring" module of the *IP SLAs Configuration Guide*.

When you are finished configuring proactive threshold monitoring for this MEP, see the "Scheduling IP SLAs Operations" section to schedule the operation.

Configuring the Sender MEP on the Source Router

Before You Begin

- Time synchronization is required between the source and destination devices in order to provide accurate one-way delay (latency) or delay-variation measurements. Configure either Precision Time Protocol (PTP) or Network Time Protocol (NTP) on both the source and destination devices.
- The receiver MEP must be configured, including proacive threshold monitoring, and scheduled before you configure the sender MEP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla operation-number**
4. **ethernet y1731 delay 1DM domain domain-name** {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **cos** *cos* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}}
5. **aggregate interval** *seconds*
6. **frame interval** *milliseconds*
7. **frame size** *bytes*
8. **history interval** *intervals-stored*
9. **owner** *owner-id*
10. **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	**ip sla operation-number** **Example:** `Router(config)# ip sla 500`	Begins configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4	**ethernet y1731 delay 1DM domain domain-name** {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **cos** *cos* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}}
	Example:
	`Router(config-ip-sla)# ethernet y1731 delay 1DM domain xxx evc yyy mpid 101 cos 3 source mpid 100`
		Begins configuring a dual-ended Ethernet delay operation and enters IP SLA Y.1731 delay configuration mode.

	Command or Action	Purpose
Step 5	**aggregate interval** *seconds* **Example:** `Router(config-sla-y1731-delay)# aggregate interval 900`	(Optional) Configures the length of time during which the performance measurements are conducted and the results stored.
Step 6	**frame interval** *milliseconds* **Example:** `Router(config-sla-y1731-delay)# frame interval 100`	(Optional) Sets the gap between successive frames.
Step 7	**frame size** *bytes* **Example:** `Router(config-sla-y1731-delay)# frame size 64`	(Optional) Sets the padding size for frames.
Step 8	**history interval** *intervals-stored* **Example:** `Router(config-sla-y1731-delay)# history interval 2`	(Optional) Sets the number of statistics distributions kept during the lifetime of an IP SLAs Ethernet operation.
Step 9	**owner** *owner-id* **Example:** `Router(config-sla-y1731-delay)# owner admin`	(Optional) Configures the owner of an IP SLAs operation.

Step 10	**end**
	Example:
	`Router(config-sla-y1731-delay)# end`
		Exits to privileged EXEC mode.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps, see the "Configuring Proactive Threshold Monitoring" module of the *IP SLAs Configuration Guide*.

When you are finished configuring proactive threshold monitoring for this MEP, see the "Scheduling IP SLAs Operations" section to schedule the operation.

Configuring a Sender MEP for a Single-Ended Ethernet Delay or Delay Variation Operation

Perform this task to configure a sender MEP on the source device.

Before You Begin

- Time synchronization is required between the source and destination devices in order to provide accurate one-way delay (latency) or delay-variation measurements. Configure either Precision Time Protocol (PTP) or Network Time Protocol (NTP) on both the source and destination devices.

Note To display information about remote (target) MEPs on destination devices, use the **show ethernet cfm maintenance-points remote** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **ethernet y1731 delay** {**DMM** | **DMMv1**} [**burst**] **domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **cos** *cos* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}}
5. **clock sync**
6. **aggregate interval** *seconds*
7. **distribution** {**delay** | **delay-variation**} **one-way** *number-of-bins boundary*[**,...,***boundary*]
8. **frame interval** *milliseconds*
9. **frame offset** *offset-value*
10. **frame size** *bytes*
11. **history interval** *intervals-stored*
12. **max-delay** *milliseconds*
13. **owner** *owner-id*
14. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config-term)# ip sla 10`	Begins configuring an IP SLAs operation and enters IP SLA configuration mode.

Step 4	**ethernet y1731 delay** {**DMM** | **DMMv1**} [**burst**] **domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **cos** *cos* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}}
	Example:
	`Device(config-ip-sla)# ethernet y1731 delay dmm domain xxx evc yyy mpid 101 cos 4 source mpid 100`
		Begins configuring a single-ended Ethernet delay operation and enters IP SLA Y.1731 delay configuration mode.
		• To configure concurrent operations, use the **DMMv1** keyword with this command. Repeat the preceding two steps to each concurrent operation, to be added to a single IP SLA operation number. Concurrent operations are supported for a given EVC, CoS, and remote MEP combination, or for multiple MEPs for a given multipoint EVC.

Step 5	**clock sync** **Example:** `Device(config-sla-y1731-delay)# clock sync`	(Optional) Indicates that the end points are synchronized and thus allows the operation to calculate one-way delay measurements.
Step 6	**aggregate interval** *seconds* **Example:** `Device(config-sla-y1731-delay)# aggregate interval 900`	(Optional) Configures the length of time during which the performance measurements are conducted and the results stored.
Step 7	**distribution** {**delay** | **delay-variation**} **one-way** *number-of-bins boundary*[,...,*boundary*] **Example:** `Device(config-sla-y1731-delay)# distribution delay-variation one-way 5 5000, 10000,15000,20000,-1`	(Optional) Specifies measurement type and configures bins for statistics distributions kept.

	Command or Action	Purpose
Step 8	**frame interval** *milliseconds* **Example:** `Device(config-sla-y1731-delay)# frame interval 100`	(Optional) Sets the gap between successive frames.
Step 9	**frame offset** *offset-value* **Example:** `Device(config-sla-y1731-delay)# frame offset 1`	(Optional) Sets value for calculating delay variation values.
Step 10	**frame size** *bytes* **Example:** `Device(config-sla-y1731-delay)# frame size 32`	(Optional) Configures padding size for frames.
Step 11	**history interval** *intervals-stored* **Example:** `Device(config-sla-y1731-delay)# history interval 2`	(Optional) Sets the number of statistics distributions kept during the lifetime of an IP SLAs Ethernet operation.
Step 12	**max-delay** *milliseconds* **Example:** `Device(config-sla-y1731-delay)# max-delay 5000`	(Optional) Sets the amount of time an MEP waits for a frame.
Step 13	**owner** *owner-id* **Example:** `Device(config-sla-y1731-delay)# owner admin`	(Optional) Configures the owner of an IP SLAs operation.

Step 14	**end**
	Example:
	`Device(config-sla-y1731-delay)# end`
		Exits to privileged EXEC mode.

What to Do Next

To add proactive threshold conditions and reactive triggering for generating traps, see the "Configuring Proactive Threshold Monitoring" module of the *IP SLAs Configuration Guide*.

When you are finished configuring proactive threshold monitoring for this operation, see the "Scheduling IP SLAs Operations" section to schedule the operation.

Configuring a Sender MEP for a Single-Ended Ethernet Frame Loss Ratio Operation

Note To display information about remote (target) MEPs on destination devices, use the **show ethernet cfm maintenance-points remote** command.

Perform this task to configure a sender MEP on the source device.

Before You Begin

- Class of Service (CoS)-level monitoring must be enabled on MEPs associated to the Ethernet frame loss operation by using the **monitor loss counter** command on the devices at both ends of the operation. See the *Cisco IOS Carrier Ethernet Command Reference* for command information. See the "Configuration Examples for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations" section for configuration information.

  Note Cisco IOS Y.1731 implementation allows monitoring of frame loss for frames on an EVC regardless of the CoS value (any CoS or Aggregate CoS cases). See the "Configuration Examples for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations" section for configuration information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** *operation-number*
4. **ethernet y1731 loss** {**LMM** | **SLM**} [**burst**] **domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **CoS** *CoS* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}}
5. **aggregate interval** *seconds*
6. **availability algorithm** {**sliding-window** | **static-window**}
7. **frame consecutive** *value*
8. **frame interval** *milliseconds*
9. **history interval** *intervals-stored*
10. **owner** *owner-id*
11. **exit**
12. **exit**
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip sla** *operation-number* **Example:** `Device(config-term)# ip sla 11`	Begins configuring an IP SLAs operation and enters IP SLA configuration mode.
Step 4	**ethernet y1731 loss** {**LMM** | **SLM**} [**burst**] **domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **CoS** *CoS* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}}	Begins configuring a single-ended Ethernet frame loss ratio operation and enters IP SLA Y.1731 loss configuration mode. • To configure concurrent operations, use the **SLM** keyword with this command. Repeat the preceding

	Command or Action	Purpose
	Example: `Device(config-ip-sla)# ethernet y1731 loss LMM domain xxx vlan 12 mpid 34 CoS 4 source mpid 23`	two steps to configure each concurrent operation to be added to a single IP SLA operation number. Concurrent operations are supported for a given EVC, CoS, and remote-MEP combination, or for multiple MEPs for a given multipoint EVC.
Step 5	**aggregate interval** *seconds* **Example:** `Device(config-sla-y1731-loss)# aggregate interval 900`	(Optional) Configures the length of time during which performance measurements are conducted and the results stored.

Step 6	**availability algorithm** {**sliding-window** | **static-window**}
	Example:
	`Device(config-sla-y1731-loss)# availability algorithm static-window`
		(Optional) Specifies availability algorithm used.

Step 7	**frame consecutive** *value* **Example:** `Device(config-sla-y1731-loss)# frame consecutive 10`	(Optional) Specifies number of consecutive measurements to be used to determine availability or unavailability status.
Step 8	**frame interval** *milliseconds* **Example:** `Device(config-sla-y1731-loss)# frame interval 100`	(Optional) Sets the gap between successive frames.
Step 9	**history interval** *intervals-stored* **Example:** `Device(config-sla-y1731-loss)# history interval 2`	(Optional) Sets the number of statistics distributions kept during the lifetime of an IP SLAs Ethernet operation.
Step 10	**owner** *owner-id* **Example:** `Device(config-sla-y1731-delay)# owner admin`	(Optional) Configures the owner of an IP SLAs operation.

	Command or Action	Purpose
Step 11	**exit** **Example:** `Device(config-sla-y1731-delay)# exit`	Exits to IP SLA configuration mode.
Step 12	**exit** **Example:** `Device(config-ip-sla)# exit`	Exits to global configuration mode.
Step 13	**exit** **Example:** `Device(config)# exit`	Exits to privileged EXEC mode.

What to Do Next

When you are finished configuring this MEP, see the "Scheduling IP SLAs Operations" section to schedule the operation.

Scheduling IP SLAs Operations

Before You Begin

- All IP Service Level Agreements (SLAs) operations to be scheduled must be already configured.
- The frequency of all operations scheduled in a multioperation group must be the same.
- The list of one or more operation ID numbers to be added to a multioperation group must be limited to a maximum of 125 characters in length, including commas (,).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter one of the following commands:
    - **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
    - **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] [**frequency** *group-operation-frequency*] [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]
4. **end**
5. **show ip sla group schedule**
6. **show ip sla configuration**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Enter one of the following commands: • **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {[*hh*:*mm*:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**] • **ip sla group schedule** *group-operation-number operation-id-numbers* {**schedule-period** *schedule-period-range* | **schedule-together**} [**ageout** *seconds*] [**frequency** *group-operation-frequency*] [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm* [:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm* [:*ss*]}]	• Configures the scheduling parameters for an individual IP SLAs operation. • Specifies an IP SLAs operation group number and the range of operation numbers for a multioperation scheduler.

	Command or Action	Purpose

	Example:
	`Device(config)# ip sla schedule 10 life forever start-time now`
	`Device(config)# ip sla schedule 10 schedule-period frequency`
	`Device(config)# ip sla group schedule 1 3,4,6-9 life forever start-time now`
	`Device(config)# ip sla schedule 1 3,4,6-9 schedule-period 50 frequency range 80-100`	

Step 4	**end**
	Example:
	`Device(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Step 5	**show ip sla group schedule** **Example:** `Device# show ip sla group schedule`	(Optional) Displays IP SLAs group schedule details.
Step 6	**show ip sla configuration** **Example:** `Device# show ip sla configuration`	(Optional) Displays IP SLAs configuration details.

Configuration Examples for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations

Example: Dual-Ended Ethernet Delay Operation

The following sample output shows the configuration, including default values, of a receiver MEP on the responder device for a dual-ended Ethernet delay or delay variation operation:

```
Device# show ip sla configuration 501

IP SLAs Infrastructure Engine-III
Entry number: 501
Owner: admin
Tag:
Operation timeout (milliseconds): 5000
Ethernet Y1731 Delay Operation
Frame Type: 1DM
Domain: xxx
```

```
ReceiveOnly: TRUE
Evc: yyy
Local Mpid: 101
CoS: 3
   Max Delay: 5000
Threshold (milliseconds): 5000
.
.
.
Statistics Parameters
  Aggregation Period: 900
  Frame offset: 1
  Distribution Delay One-Way:
   Number of Bins 10
   Bin Boundaries: 5000,10000,15000,20000,25000,30000,35000,40000,45000,-1
  Distribution Delay-Variation One-Way:
   Number of Bins 10
   Bin Boundaries: 5000,10000,15000,20000,25000,30000,35000,40000,45000,-1
History
  Number of intervals: 2
```

The following sample output shows the configuration, including default values, of the sender MEP for a dual-ended IP SLAs Ethernet delay or delay variation operation:

```
Device# show ip sla configuration 500

IP SLAs Infrastructure Engine-III
Entry number: 500
Owner:
Tag:
Operation timeout (milliseconds): 5000
Ethernet Y1731 Delay Operation
Frame Type: 1DM
Domain: yyy
ReceiveOnly: FALSE
Evc: xxx
Target Mpid: 101
Source Mpid: 100
CoS: 3
   Request size (Padding portion): 64
   Frame Interval: 1000
Threshold (milliseconds): 5000
.
.
.
Statistics Parameters
  Aggregation Period: 900
  Frame offset: 1
History
  Number of intervals: 22
```

Example: Frame Delay and Frame Delay Variation Measurement Configuration

The following sample output shows the performance monitoring session summary:

```
Device# show ethernet cfm pm session summary

Number of Configured Session : 2
Number of Active Session: 2
Number of Inactive Session: 0
```

The following sample output shows the active performance monitoring session:

```
Device# show ethernet cfm pm session active

Display of Active Session
--------------------------------------------------------------------------------
EPM-ID   SLA-ID    Lvl/Type/ID/Cos/Dir     Src-Mac-address Dst-Mac-address
--------------------------------------------------------------------------------
 0     10           3/BD-V/10/2/Down     d0c2.8216.c9d7  d0c2.8216.27a3
 1     11           3/BD-V/10/3/Down     d0c2.8216.c9d7  d0c2.8216.27a3
Total number of Active Session: 2
Device# show ethernet cfm pm session db 0

----------------------------------------------------------------------------
      TX Time FWD                   RX Time FWD
      TX Time BWD                   RX Time BWD                 Frame Delay
      Sec:nSec                      Sec:nSec                     Sec:nSec
----------------------------------------------------------------------------
Session ID: 0
****************************************************************************
       234:526163572                  245:305791416
       245:306761904                  234:527134653            0:593
****************************************************************************
       235:528900628                  246:308528744
       246:309452848                  235:529825333            0:601
****************************************************************************
       236:528882716                  247:308511128
       247:309450224                  236:529822413            0:601
****************************************************************************
       237:526578788                  248:306207432
       248:307157936                  237:527529885            0:593
****************************************************************************
       238:527052156                  249:306681064
       249:307588016                  238:527959717            0:609
****************************************************************************
       239:526625044                  250:306254200
       250:307091888                  239:527463325            0:593
****************************************************************************
       240:528243204                  251:307872648
       251:308856880                  240:529228021            0:585
```

Example: Sender MEP for a Single-Ended Ethernet Delay Operation

The following sample output shows the configuration, including default values, of the sender MEP for a single-ended IP SLAs Ethernet delay operation:

```
Router# show ip sla configuration 10

IP SLAs Infrastructure Engine-III
Entry number: 10
Owner:
Tag:
Operation timeout (milliseconds): 5000
Ethernet Y1731 Delay Operation
Frame Type: DMM
Domain: xxx
Vlan: yyy
Target Mpid: 101
Source Mpid: 100
CoS: 4
   Max Delay: 5000
   Request size (Padding portion): 64
   Frame Interval: 1000
   Clock: Not In Sync
Threshold (milliseconds): 5000
.
.
.
Statistics Parameters
  Aggregation Period: 900
  Frame offset: 1
  Distribution Delay Two-Way:
   Number of Bins 10
   Bin Boundaries: 5000,10000,15000,20000,25000,30000,35000,40000,45000,-1
  Distribution Delay-Variation Two-Way:
   Number of Bins 10
   Bin Boundaries: 5000,10000,15000,20000,25000,30000,35000,40000,45000,-1
History
```

```
  Number of intervals: 2
```

Example: Sender MEP for a Single-Ended Ethernet Frame Loss Operation

The following output shows the configuration, including default values, of the sender MEP in a basic single-ended IP SLAs Ethernet frame loss ratio operation with a start-time of now:

```
Router# show ip sla configuration 11

IP SLAs Infrastructure Engine-III
Entry number: 11
Owner:
Tag:
Operation timeout (milliseconds): 5000
Ethernet Y1731 Loss Operation
Frame Type: LMM
Domain: xxx
Vlan: 12
Target Mpid: 34
Source Mpid: 23
CoS: 4
   Request size (Padding portion): 0
   Frame Interval: 1000
Schedule:
   Operation frequency (seconds): 60  (not considered if randomly scheduled)
   Next Scheduled Start Time: Start Time already passed
   Group Scheduled : FALSE
   Randomly Scheduled : FALSE
   Life (seconds): 3600
   Entry Ageout (seconds): never
   Recurring (Starting Everyday): FALSE
   Status of entry (SNMP RowStatus): ActiveThreshold (milliseconds): 5000
Statistics Parameters
  Aggregation Period: 900
  Frame consecutive: 10
  Availability algorithm: static-window
History
  Number of intervals: 2
```

Additional References for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS Carrier Ethernet commands	Cisco IOS Carrier Ethernet Command Reference
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference

Related Topic	Document Title
Ethernet CFM	"Configuring Ethernet Connectivity Fault Management in a Service Provider Network" module of the *Cisco IOS Carrier Ethernet Configuration Guide*
Network Time Protocol (NTP)	"Configuring NTP" module of the *Cisco IOS Network Management Configuration Guide*
Proactive threshold monitoring for Cisco IOS IP SLAs	"Configuring Proactive Threshold Monitoring of IP SLAs Operations" module of the *Cisco IOS IP SLAs Configuration Guide*

Standards and RFCs

Standard/RFC	Title
ITU-T Y.1731	*OAM functions and mechanisms for Ethernet-based networks*
No specific RFCs are supported by the features in this document.	--

MIBs

MIB	MIBs Link
• CISCO-IPSLA-ETHERNET-MIB • CISCO-RTTMON-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

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 IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations

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 29: Feature Information for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731)

Feature Name	Releases	Feature Information
IP SLA Support for ETH-SLM (Ethernet Synthetic Loss Measurement in Y1731)		Y.1731 Performance Monitoring (PM) provides a standard Ethernet PM function that includes measurement of Ethernet frame delay, frame delay variation, frame loss, and frame throughput measurements specified by the ITU-T Y-1731 standard and interpreted by the Metro Ethernet Forum (MEF) standards group.
Y1731 MIB Support through existing IPSLA MIBs		Support was added for reporting threshold events and collecting performance statistics for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) operations using SNMP.

CHAPTER 22

IPSLA Y1731 On-Demand and Concurrent Operations

This module describes how to configure the IPSLA Y1731 SLM Feature Enhancements feature for enabling real-time Ethernet service troubleshooting for users without configuration privileges. This feature supports on-demand Synthetic Loss Measurement (SLM) operations that can be run by issuing a single command in privileged EXEC mode.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 ITU-T Y.1731 Operations

IEEE-compliant Connectivity Fault Management (CFM) must be configured and enabled for Y.1731 performance monitoring to function.

Restrictions for IP SLAs Y.1731 On-Demand Operations

- SNMP is not supported for reporting threshold events or collecting performance statistics for on-demand operations.
- On-demand operation statistics are not stored and are not supported by the statistic history and aggregation functions.

Information About IP SLAs Y.1731 On-Demand and Concurrent Operations

IPSLA Y1731 SLM Feature Enhancements

On-demand IP SLAs Synthetic Loss Measurement (SLM) operations, in the IPSLA Y1731 SLM Feature Enhancements feature, enable users without configuration access to perform real-time troubleshooting of Ethernet services. There are two operational modes for on-demand operations: direct mode that creates and runs an operation immediately and referenced mode that starts and runs a previously configured operation.

- In the direct mode, a single command can be used to create multiple pseudo operations for a range of class of service (CoS) values to be run, in the background, immediately. A single command in privileged EXEC mode can be used to specify frame size, interval, frequency, and duration for the direct on-demand operation. Direct on-demand operations start and run immediately after the command is issued.
- In the referenced mode, you can start one or more already-configured operations for different destinations, or for the same destination, with different CoS values. Issuing the privileged EXEC command creates a pseudo version of a proactive operation that starts and runs in the background, even while the proactive operation is running.
- Once an on-demand operation is completed, statistical output is displayed on the console. On-demand operation statistics are not stored and are not supported by the statistic history and aggregation functions.
- After an on-demand operation is completed, and the statistics handled, the direct and referenced on-demand operation is deleted. The proactive operations are not deleted and continue to be available to be run in referenced mode, again.

A concurrent operation consists of a group of operations, all configured with the same operation ID number, that run concurrently. Concurrent operations are supported for a given Ethernet Virtual Circuit (EVC), CoS, and remote Maintenance End Point (MEP) combination, or for multiple MEPs for a given multipoint EVC, for delay or loss measurements. A new keyword was added to the appropriate commands to specify that concurrent Ethernet frame Delay Measurement (ETH-DM) synthetic frames are sent during the operation.

The IPSLA Y.1731 SLM Feature Enhancements feature also supports burst mode for concurrent operations, one-way dual-ended, and single-ended delay and delay variation operations, as well as for single-ended loss operations. A new keyword was added to the appropriate commands to support bursts of PDU transmission during an aggregation interval. The maximum number of services monitored is 50 every 30 minutes, with an average of 25 services every 2 hours.

How to Configure IP SLAs Y.1731 On-Demand and Concurrent Operations

Configuring a Direct On-Demand Operation on a Sender MEP

Before You Begin

Class of Service (CoS)-level monitoring must be enabled on MEPs associated to the Ethernet frame loss operation by using the **monitor loss counter** command on the devices at both ends of the operation. See the *Cisco IOS Carrier Ethernet Command Reference* for command information. See the “Configuration Examples for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations” section for configuration information.

Note The Cisco IOS Y.1731 implementation allows monitoring of frame loss on an EVC regardless of the CoS value (any CoS or aggregate CoS cases). See the “Configuration Examples for IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations” section for configuration information.

SUMMARY STEPS

1. **enable**
2. **ip sla on-demand ethernet** {**DMMv1** | **SLM**} **domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **cos** *cos* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}} {**continuous** [**interval** *milliseconds*] | **burst** [**interval** *milliseconds*] [**number** *number-of-frames*] [**frequency** *seconds*]} [**size** *bytes*] **aggregation** *seconds* {**duration** *seconds* | **max** *number-of-packets*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ip sla on-demand ethernet** {**DMMv1** | **SLM**} **domain** *domain-name* {**evc** *evc-id* | **vlan** *vlan-id*} {**mpid** *target-mp-id* | **mac-address** *target-address*} **cos** *cos* {**source** {**mpid** *source-mp-id* | **mac-address** *source-address*}} {**continuous** [**interval** *milliseconds*] | **burst** [**interval** *milliseconds*] [**number** *number-of-frames*] [**frequency** *seconds*]} [**size** *bytes*] **aggregation** *seconds* {**duration** *seconds* | **max** *number-of-packets*}	Creates and runs an on-demand operation in direct mode. • To create and run concurrent on-demand operations, configure this command using the **DMMv1** keyword. • Statistical output is posted on the console after the operation is finished. • Repeat this step for each on-demand operation to be run.

	Command or Action	Purpose
	Example: `Device# ip sla on-demand ethernet SLM domain xxx vlan 12 mpid 34 cos 4 source mpid 23 continuous aggregation 10 duration 60`	• After an on-demand operation is finished and the statistics handled, the operation is deleted.

Configuring a Referenced On-Demand Operation on a Sender MEP

Note After an on-demand operation is finished and the statistics handled, the on-demand version of the operation is deleted.

Before You Begin

- Single-ended and concurrent Ethernet delay, or delay variation, and frame loss operations to be referenced must be configured. See the “Configuring IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations” module of the *IP SLAs Configuration Guide*.

SUMMARY STEPS

1. **enable**
2. **ip sla on-demand ethernet** [**dmmv1** | **slm**] *operation-number* {**duration** *seconds* | **max** *number-of-packets*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**ip sla on-demand ethernet** [**dmmv1** | **slm**] *operation-number* {**duration** *seconds* | **max** *number-of-packets* **Example:** `Device# ip sla on-demand ethernet slm 11 duration 38`	Creates and runs a pseudo operation of the operation being referenced, in the background. • Statistical output is posted on the console after the operation is finished. • Repeat this step for each on-demand operation to be run.

Configuring an IP SLAs Y.1731 Concurrent Operation on a Sender MEP

To configure concurrent Ethernet delay, delay variation, and frame loss operations, see the “Configuring IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations” module of the *IP SLAs Configuration Guide.*

Configuration Examples for IP SLAs Y.1731 On-Demand and Concurrent Operations

Example: On-Demand Operation in Direct Mode

```
Device# ip sla on-demand ethernet SLM domain xxx vlan 10 mpid 3 cos 1 source mpid 1 continuous
 aggregation 35 duration 38

Loss Statistics for Y1731 Operation 2984884426
Type of operation: Y1731 Loss Measurement
Latest operation start time: *20:17:41.535 PST Wed May 16 2012
Latest operation return code: OK
Distribution Statistics:


Interval 1
 Start time:  *20:17:41.535 PST Wed May 16 2012
 End time:  *20:18:16.535 PST Wed May 16 2012
 Number of measurements initiated: 35
 Number of measurements completed: 35
 Flag: OK

Forward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps forward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
Backward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps backward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
Loss Statistics for Y1731 Operation 2984884426
Type of operation: Y1731 Loss Measurement
Latest operation start time: *20:17:41.535 PST Wed May 16 2012
Latest operation return code: OK
Distribution Statistics:


Interval 1
 Start time:  *20:17:41.535 PST Wed May 16 2012
 End time:  *20:18:16.535 PST Wed May 16 2012
 Number of measurements initiated: 35
 Number of measurements completed: 35
 Flag: OK

Forward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps forward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
Backward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps backward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
```

Example: On-Demand Operation in Referenced Mode

```
Device(config)# ip sla 11
Device(config-ip-sla)# ethernet y1731 loss SLM domain xxx vlan 10 mpid 3 cos 1 source mpid
 1
Device(config-sla-y1731-loss)# end
Device# ip sla on-demand ethernet slm 11 duration 38

Loss Statistics for Y1731 Operation 2984884426
Type of operation: Y1731 Loss Measurement
Latest operation start time: *20:17:41.535 PST Wed May 16 2012
Latest operation return code: OK
Distribution Statistics:


Interval 1
 Start time:  *20:17:41.535 PST Wed May 16 2012
 End time:  *20:18:16.535 PST Wed May 16 2012
 Number of measurements initiated: 35
 Number of measurements completed: 35
 Flag: OK

Forward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps forward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
Backward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps backward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
Loss Statistics for Y1731 Operation 2984884426
Type of operation: Y1731 Loss Measurement
Latest operation start time: *20:17:41.535 PST Wed May 16 2012
Latest operation return code: OK
Distribution Statistics:


Interval 1
 Start time:  *20:17:41.535 PST Wed May 16 2012
 End time:  *20:18:16.535 PST Wed May 16 2012
 Number of measurements initiated: 35
 Number of measurements completed: 35
 Flag: OK

Forward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps forward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
Backward
  Number of Observations 3
  Available indicators: 0
  Unavailable indicators: 3
  Tx frame count: 30
  Rx frame count: 30
    Min/Avg/Max - (FLR % ): 0:9/000.00%/0:9
  Cumulative - (FLR % ): 000.00%
  Timestamps backward:
    Min - *20:18:10.586 PST Wed May 16 2012
    Max - *20:18:10.586 PST Wed May 16 2012
```

Additional References for IP SLAs Y.1731 On-Demand and Concurrent Operations

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS Carrier Ethernet commands	Cisco IOS Carrier Ethernet Command Reference
Cisco IOS IP SLAs commands	Cisco IOS IP SLAs Command Reference

Related Topic	Document Title
Ethernet CFM for ITU-T Y.1731	"ITU-T Y.1731 Performance Monitoring in a Service Provider Network" module of the *Carrier Ethernet Configuration Guide*
Ethernet operations	"Configuring IP SLAs Metro-Ethernet 3.0 (ITU-T Y.1731) Operations" module of the *IP SLAs Configuration Guide*
Network Time Protocol (NTP)	"Configuring NTP" module of the *Network Management Configuration Guide*

Standards and RFCs

Standard/RFC	Title
ITU-T Y.1731	*OAM functions and mechanisms for Ethernet-based networks*

MIBs

MIB	MIBs Link
• CISCO-IPSLA-ETHERNET-MIB • CISCO-RTTMON-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

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 IP SLAs Y.1731 On-Demand and Concurrent Operations

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 30: Feature Information for IP SLAs Y.1731 On-Demand and Concurrent Operations

Feature Name	Releases	Feature Information
IPSLA Y1731 SLM Feature Enhancements		This feature enhancement allows you to run on-demand Synthetic Loss Measurement (SLM) operations, independent from previously scheduled operations, for the purpose of troubleshooting Etherent services in your network. The following commands were introduced or modified: **ethernet y1731 delay**, **ethernet y1737 loss**, **ip sla on-demand ethernet**.