Ethernet Connectivity Fault Management

First Published: June 19, 2006
Last Updated: February 18, 2009

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

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

Finding Feature Information in This Module

Your Cisco IOS software release may not support all of the features documented in this module. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the "Feature Information for Ethernet Connectivity Fault Management" section on page 74.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Ethernet Connectivity Fault Management

Business Requirements

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

Restrictions for Ethernet Connectivity Fault Management

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

- For Policy Feature Card (PFC)-based EoMPLS, which uses a Cisco Catalyst LAN card as the MPLS uplink port, a CFM packet can be transparently passed via an EoMPLS pseudowire like regular data packets. The EoMPLS endpoint interface, however, cannot be a MEP or a MIP, although a CFM MEP or MIP can be supported on regular Layer 2 switchport interfaces.
- For switched virtual interface (SVI)-based EoMPLS or virtual private LAN service (VPLS), both of which use a WAN card as an MPLS uplink port, CFM packets can be transparently passed via an EoMPLS pseudowire only if CFM is globally disabled on the same system.

Information About Ethernet Connectivity Fault Management

Before you set up Ethernet CFM, you should understand the following concepts:

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 customer edge (CE) to CE. Per service instance means per VLAN.

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 uPE and CE and relies on CFM for reporting status of the metro-Ethernet network to the CE.

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

Benefits of Ethernet CFM

Ethernet CFM provides the following benefits:

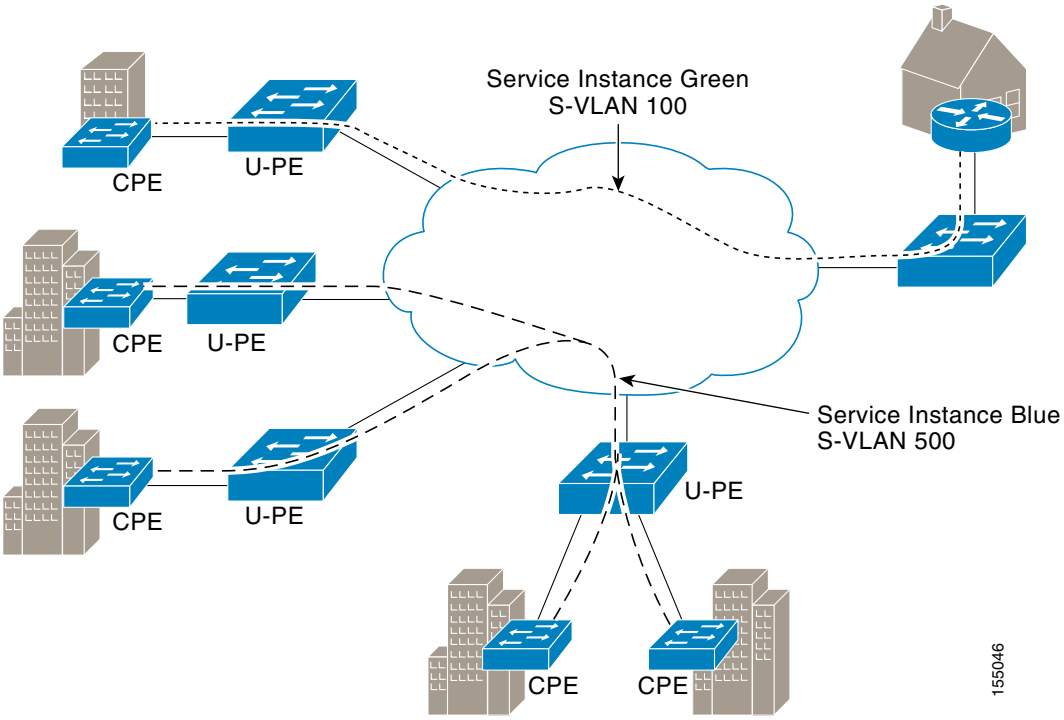
- End-to-end service-level OAM technology
- Reduced operating expense for service provider Ethernet networks
- Competitive advantage for service providers

- 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. Figure 1 shows two customer service instances. Service Instance Green is point to point; Service Instance Blue is multipoint to multipoint.

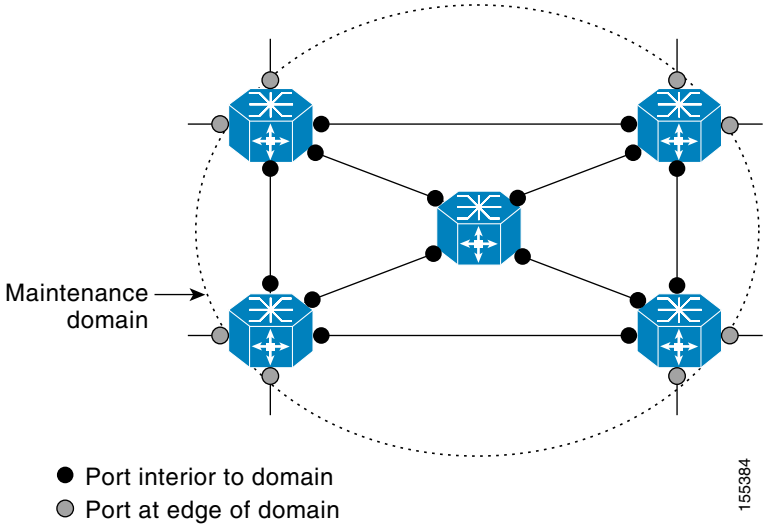
Figure 1 Customer Service Instances



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. Figure 2 illustrates a typical maintenance domain.

Figure 2 Ethernet CFM Maintenance Domain



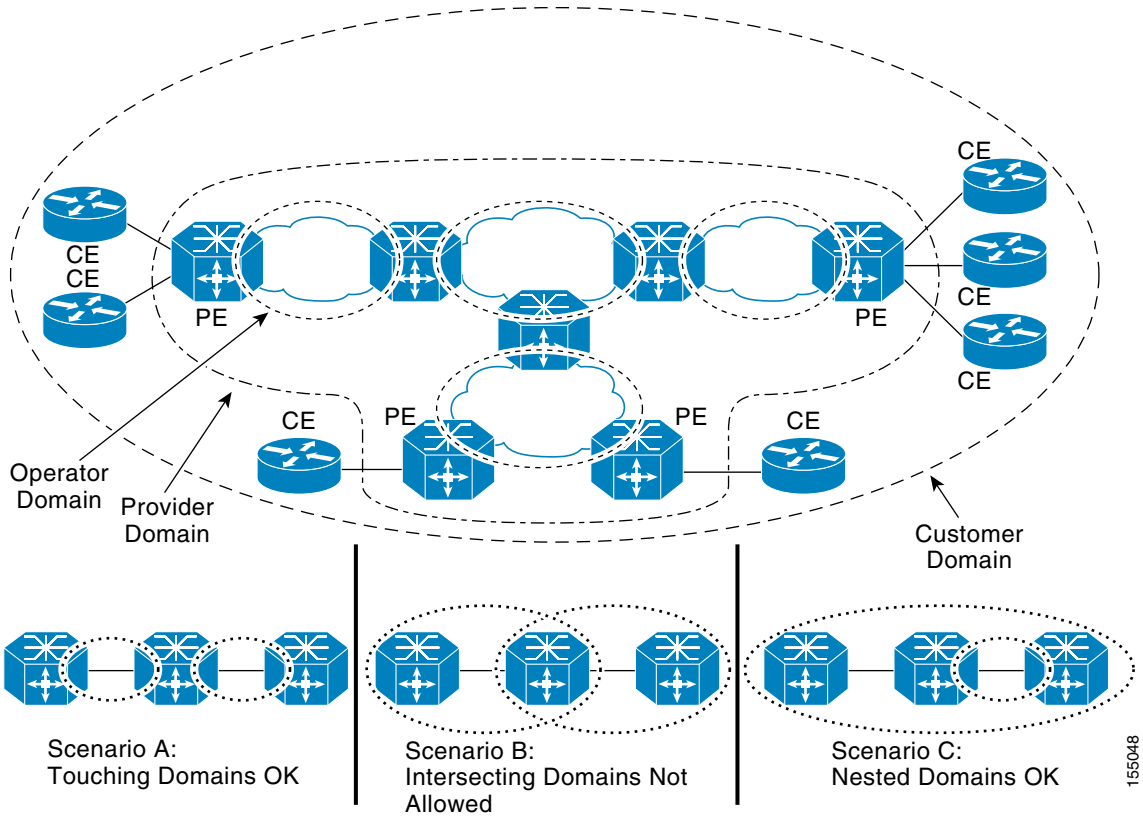
A unique maintenance level in the range of 0 to 7 is assigned to each domain by a network administrator. Levels and domain names are useful for defining the hierarchical relationship that exists among domains. The hierarchical relationship of domains parallels the structure of customer, service provider, 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. Figure 3 illustrates a hierarchy of operator, service provider, and customer domains and also illustrates touching, intersecting, and nested domains.

Figure 3 Ethernet CFM Maintenance Domain Hierarchy



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

MEPs have the following characteristics:

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

Inward Facing MEPs

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

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

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

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

Outward Facing MEPs

Outward facing means that the MEP communicates through the wire. Outward facing MEPs use the port MAC address, not the Bridge-Brain MAC address used by inward facing MEPs. Outward facing MEPs can be configured only on routed ports. A MIP configuration at a level higher than the level of the outward facing MEP is not required.

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 relay function side.
- 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. Not applicable to routed ports.
- If the port on which the outward MEP is configured is blocked by Spanning-Tree Protocol, the MEP can still transmit and receive CFM messages via the wire.

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.

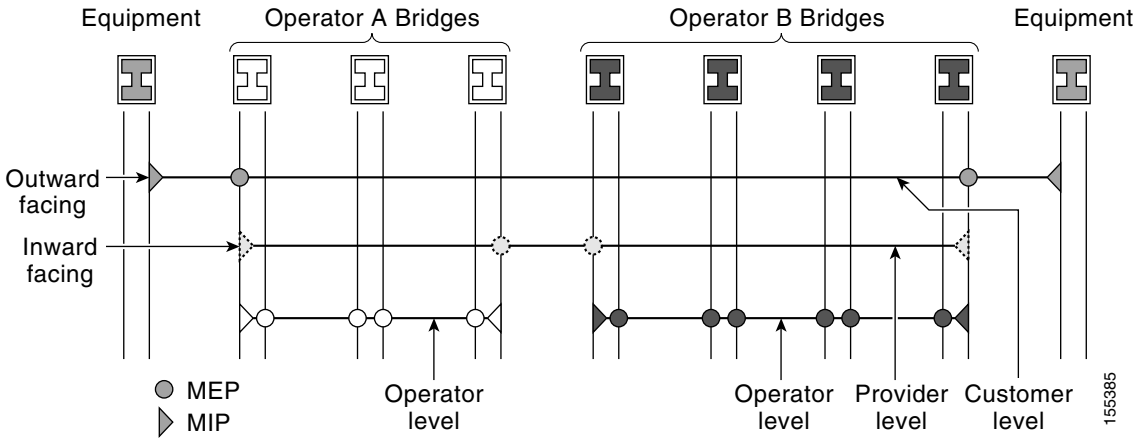
- All CFM frames at a higher level are forwarded, independent of whether they arrive from the wire or relay function.
- Passive points, respond only when triggered by CFM traceroute and loopback messages.
- Bridge-Brain MAC addresses are used.

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.

Figure 4 illustrates MEPs and MIPs at the operator, service provider, and customer levels.

Figure 4 CFM MEPs and MIPs on Customer and Service Provider Equipment, Operator Devices



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 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 and S-VLAN.

CFM CCMs have the following characteristics:

- Transmitted at a configurable periodic interval by MEPs. The interval can be from 10 seconds to 65535 seconds, the default is 30.
- Contain a configurable hold-time value to indicate to the receiver the validity of the message. The default is 2.5 times the transmit interval.
- Catalogued by MIPs at the same maintenance level.
- Terminated by remote MEPs at the same maintenance level.
- Unidirectional and do not solicit a response.
- Carry the status of the port on which the MEP is configured.

Loopback Messages

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

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

Traceroute Messages

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

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

Cross-Check Function

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

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

SNMP Traps

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

CC Traps

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

Cross-Check Traps

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

Ethernet CFM and Ethernet OAM Interaction

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

- Ethernet Virtual Circuit, page 10
- OAM Manager, page 10

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

OAM Manager

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

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

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

How to Set Up Ethernet Connectivity Fault Management

To set up Ethernet CFM, perform the following tasks:

- Designing CFM Domains, page 11
- Configuring Ethernet CFM, page 13

Designing CFM Domains

Perform the steps described in this section to design CFM domains for Ethernet CFM functionality.

Prerequisites

- 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

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.

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

Step 1 Determine operator level MIPs.

a. Starting at lowest operator level domain, assign a MIP at every interface internal to the operator network to be visible to CFM.

b. Proceed to next higher operator level and assign MIPs.

c. Verify that every port that has a MIP at a lower level does not have maintenance points at a higher level.

d. Repeat steps a through d until all operator MIPs are determined.

Step 2 Determine operator level MEPs.

a. Starting at the lowest operator level domain, assign a MEP at every user network interface (UNI) that is part of a service instance.

b. Assign a MEP at the network to network interface (NNI) between operators, if there is more than one operator.

c. 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.

a. Starting at the lowest service provider level domain, assign service provider MIPs at the NNI between operators (if more than one).

b. 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.

a. Starting at the lowest service provider level domain, assign a MEP at every UNI that is part of a service instance.

b. Proceed to next higher service provider level and assign MEPs.

   A port with a MIP at a lower level cannot have maintenance points at a higher level. A port with a MEP at a lower level should have either a MIP or a MEP at a higher level.

Step 5 Determine customer MIPs.

Customer MIPs are allowed only on the UNIs at the uPEs if the service provider allows the customer to run CFM. Otherwise, the service provider can configure Cisco IOS devices to block CFM frames.

a. Configure a MIP on every uPE, at the UNI port, in the customer maintenance domain. Ensure the MIPs are at a maintenance level that is at least one higher than the highest level service provider domain.

Step 6 Determine customer MEPs.

Customer MEPs are on customer equipment.

a. 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

Figure 5 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.

Figure 5 Configuring Domains for Ethernet CFM

What to Do Next

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

Configuring Ethernet CFM

Configuring Ethernet CFM consists of the following tasks:

- Provisioning the Network, page 13 (required)
- Provisioning Service, page 34 (required)
- Configuring and Enabling the Cross-Check Function, page 56 (optional)

Provisioning the Network

Perform this task to prepare the network for Ethernet CFM.

SUMMARY STEPS

CE-A

1. **enable**
2. **configure terminal**

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

DETAILED STEPS

	Command or Action	Purpose
	CE-A	
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **direction outward** **Example:** `Router(config)# ethernet cfm domain Customer level 7 direction outward`	Defines an outward CFM maintenance domain named Customer at level 7 and puts the CLI in Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 60 minutes.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 7	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 8	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table to 200 messages.
Step 9	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the time that CFM traceroute cache entries are retained to 60 minutes.

	Command or Action	Purpose
Step 10	**ethernet cfm cc** {**level any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} {**vlan any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num_msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the following parameters for CCMs: • All maintenance levels are to be configured. • All VLANs are to be configured. • The time between CCM transmissions is 20 seconds. • The maximum number of CCMs that can be missed before a MEP is declared down is 3.
Step 11	**snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.
Step 12	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown** | **mep-missing** | **service-up**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up`	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.
Step 13	**Ctrl-Z** **Example:** `Router(config)# Ctrl Z`	Returns the CLI to privileged EXEC mode.

SUMMARY STEPS

U-PE A

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **ethernet cfm domain** *domain-name* **level** *level-id*
5. **mep archive-hold-time** *minutes*
6. **ethernet cfm domain** *domain-name* **level** *level-id*
7. **mep archive-hold-time** *minutes*
8. **exit**
9. **ethernet cfm enable**
10. **ethernet cfm traceroute cache**
11. **ethernet cfm traceroute cache size** *entries*
12. **ethernet cfm traceroute cache hold-time** *minutes*

13. **interface** *interface-type interface-id*
14. **ethernet cfm mip level** *level-id*
15. **exit**
16. **ethernet cfm cc** {**level any** | *level-id* | ,*level-id* | *level-id-level-id* | ,*level-id-level-id*} {**vlan any** | *vlan-id* | ,*vlan-id* | *vlan-id-vlan-id* | ,*vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num_msgs*]
17. **snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**]
18. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown** | **mep-missing** | **service-up**]
19. **Ctrl-Z**

DETAILED STEPS

	Command or Action	Purpose
	U-PE A	
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 named Customer at level 7 and puts the CLI in Ethernet CFM configuration mode.
Step 4	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain named ServiceProvider at level 4.
Step 5	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 60 minutes.
Step 6	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain OperatorA level 1`	Defines a domain named OperatorA at level 1.
Step 7	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 65 minutes.

	Command or Action	Purpose
Step 8	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 9	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 10	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 11	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table to 200 messages.
Step 12	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the time that CFM traceroute cache entries are retained to 60 minutes.
Step 13	**interface** *interface-type interface-id* **Example:** `Router(config)# interface gigabitethernet4/2`	Configures interface Gigabit Ethernet 4/2 and enters interface configuration mode.
Step 14	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP at maintenance level 1 on the interface.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.
Step 16	**ethernet cfm cc** {**level any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} {**vlan any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num_msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the following parameters for CCMs: • All maintenance levels are to be configured. • All VLANs are to be configured. • The time between CCM transmissions is 20 seconds. • The maximum number of CCMs that can be missed before a MEP is declared down is 3.

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

SUMMARY STEPS

PE-AGG A

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

DETAILED STEPS

	Command or Action	Purpose
	PE-AGG A	
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 named OperatorA at level 1 and puts the CLI in Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 65 minutes.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 7	**interface** *interface-type interface-id* **Example:** `Router(config)# interface gigabitethernet3/1`	Configures interface Gigabit Ethernet 3/1 and puts the CLI in interface configuration mode.
Step 8	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP at maintenance level 1 on the interface.
Step 9	**interface** *interface-type interface-id* **Example:** `Router(config-if)# interface gigabitethernet4/1`	Configures interface Gigabit Ethernet 4/1.

	Command or Action	Purpose
Step 10	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP at maintenance level 1 on the interface.
Step 11	**Ctrl-Z** **Example:** `Router(config-if)# Ctrl Z`	Returns the CLI to privileged EXEC mode.

SUMMARY STEPS

N-PE A

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

DETAILED STEPS

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

	Command or Action	Purpose
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 to 200 messages.
Step 11	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the time that CFM traceroute cache entries are retained to 60 minutes.
Step 12	**interface** *interface-type interface-id* **Example:** `Router(config)# interface gigabitethernet3/0`	Configures interface Gigabit Ethernet 3/0 and puts the CLI in interface configuration mode.
Step 13	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP at maintenance level 1 on the interface.
Step 14	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.
Step 15	**ethernet cfm cc** {**level any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} {**vlan any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num_msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the following parameters for CCMs: • All maintenance levels are to be configured. • All VLANs are to be configured. • The time between CCM transmissions is 20 seconds. • The maximum number of CCMs that can be missed before a MEP is declared down is 3.
Step 16	**snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.

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

SUMMARY STEPS

U-PE B

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

DETAILED STEPS

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

	Command or Action	Purpose
Step 10	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 11	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table to 200 messages.
Step 12	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the time that CFM traceroute cache entries are retained to 60 minutes.
Step 13	**interface** *interface-type interface-id* **Example:** `Router(config)# interface gigabitethernet2/0`	Configures interface Gigabit Ethernet 2/0 and puts the CLI in interface configuration mode.
Step 14	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at maintenance level 2 on the interface.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.
Step 16	**ethernet cfm cc** {**level any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} {**vlan any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num_msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the following parameters for CCMs: • All maintenance levels are to be configured. • All VLANs are to be configured. • The time between CCM transmissions is 20 seconds. • The maximum number of CCMs that can be missed before a MEP is declared down is 3.
Step 17	**snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.

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

SUMMARY STEPS

PE-AGG B

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

DETAILED STEPS

	Command or Action	Purpose
	PE-AGG B	
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 named OperatorB at level 2 and puts the CLI in Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 65 minutes.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 7	**interface** *interface-type interface-id* **Example:** `Router(config)# interface gigabitethernet1/1`	Configures interface Gigabit Ethernet 1/1 and puts the CLI in interface configuration mode.
Step 8	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at maintenance level 2 on the interface.
Step 9	**interface** *interface-type interface-id* **Example:** `Router(config)# interface gigabitethernet2/1`	Configures interface Gigabit Ethernet 2/1.

	Command or Action	Purpose
Step 10	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at maintenance level 2 on the interface.
Step 11	**Ctrl-Z** **Example:** `Router(config-if)# Ctrl Z`	Returns the CLI to privileged EXEC mode.

SUMMARY STEPS

N-PE B

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

DETAILED STEPS

	Command or Action	Purpose
	N-PE B	
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 cc** {**level any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} {**vlan any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num_msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Puts the CLI in Ethernet CFM configuration mode and sets the following parameters for CCMs: • All maintenance levels are to be configured. • All VLANs are to be configured. • The time between CCM transmissions is 20 seconds. • The maximum number of CCMs that can be missed before a MEP is declared down is 3.
Step 4	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain named ServiceProvider at level 4 and puts the CLI in Ethernet CFM configuration mode.
Step 5	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 60 minutes.
Step 6	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain OperatorB level 2`	Defines a domain named OperatorB at level 2.
Step 7	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 65 minutes.
Step 8	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.

	Command or Action	Purpose
Step 9	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 10	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 11	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table to 200 messages.
Step 12	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the time that CFM traceroute cache entries are retained to 60 minutes.
Step 13	**interface** *interface-type interface-id* **Example:** `Router(config)# interface gigabitethernet1/2`	Configures interface Gigabit Ethernet 1/2 and puts the CLI in interface configuration mode.
Step 14	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at maintenance level 2 on the interface.
Step 15	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.
Step 16	**snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.

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

SUMMARY STEPS

CE-B

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

DETAILED STEPS

	Command or Action	Purpose
	CE-B	
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain Customer level 7 direction outward`	Defines an outward CFM maintenance domain named Customer at level 7 and puts the CLI in Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 60 minutes.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 7	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 8	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table to 200 messages.
Step 9	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the time that CFM traceroute cache entries are retained to 60 minutes.

	Command or Action	Purpose
Step 10	**ethernet cfm cc** {**level any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} {**vlan any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num-msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the following parameters for CCMs: • All maintenance levels are to be configured. • All VLANs are to be configured. • The time between CCM transmissions is 20 seconds. • The maximum number of CCMs that can be missed before a MEP is declared down is 3.
Step 11	**snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.
Step 12	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown** | **mep-missing** | **service-up**] **Example:** `Router(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up`	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.
Step 13	**Ctrl-Z** **Example:** `Router(config)# Ctrl Z`	Returns the CLI to privileged EXEC mode.

Provisioning Service

Perform the following 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, page 56.

SUMMARY STEPS

CE-A

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
4. **mep archive-hold-time** *minutes*
5. **service** *csi-id* **vlan** *vlan-id*
6. **exit**
7. **ethernet cfm enable**
8. **ethernet cfm traceroute cache**

9. **ethernet cfm traceroute cache size** *entries*
10. **ethernet cfm traceroute cache hold-time** *minutes*
11. **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*}
12. **Ctrl-Z**

DETAILED STEPS

	Command or Action	Purpose
	CE-A	
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain Customer level 7 direction outward`	Defines an outward CFM maintenance domain named Customer at level 7 and puts the CLI in configuration Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 60 minutes.
Step 5	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1 vlan 100`	Sets a universally unique ID of MetroCustomer1 on VLAN 100 for a CSI within the maintenance domain ServiceProvider.
Step 6	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 7	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 8	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.

	Command or Action	Purpose
Step 9	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table to 200 messages.
Step 10	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the time that CFM traceroute cache entries are retained to 60 minutes.
Step 11	**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 a MEP on the interface at maintenance level 7, in outward domain Customer, with an ID of 701, and on VLAN 100.
Step 12	**Ctrl-Z** **Example:** `Router(config)# Ctrl Z`	Returns the CLI to privileged EXEC mode.

SUMMARY STEPS

U-PE A

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **ethernet cfm domain** *domain-name* **level** *level-id*
5. **mep archive-hold-time** *minutes*
6. **service** *csi-id* **vlan** *vlan-id*
7. **ethernet cfm domain** *domain-name* **level** *level-id*
8. **mep archive-hold-time** *minutes*
9. **service** *csi-id* **vlan** *vlan-id*
10. **exit**
11. **ethernet cfm enable**
12. **ethernet cfm traceroute cache**
13. **ethernet cfm traceroute cache size** *entries*
14. **ethernet cfm traceroute cache hold-time** *minutes*
15. **interface** *interface-type interface-id*
16. **ethernet cfm mip level** *level-id*

17. **ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*}
18. **ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*}
19. **interface** *interface-type interface-id*
20. **ethernet cfm mip level** *level-id*
21. **ethernet cfm cc enable level** {**any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*}
22. **ethernet cfm cc enable level** {**any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*}
23. **ethernet cfm cc** {**level any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} {**vlan any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num_msgs*]
24. **Ctrl-Z**

DETAILED STEPS

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

	Command or Action	Purpose
Step 7	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain OperatorA level 1`	Defines a domain named OperatorA at level 1.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 65 minutes.
Step 9	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1OpA vlan 100`	Sets a universally unique ID of MetroCustomer1OpA on VLAN 100 for a CSI within the maintenance domain OperatorA.
Step 10	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 11	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 12	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 13	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table to 200 messages.
Step 14	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the time that CFM traceroute cache entries are retained to 60 minutes.
Step 15	**interface** *interface-type interface-id* **Example:** `Router(config)# interface gigabitethernet3/2`	Configures interface Gigabit Ethernet 3/2 and puts the CLI in interface configuration mode.
Step 16	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 7`	Provisions a MIP at maintenance level 7 on the interface.

	Command or Action	Purpose
Step 17	**ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep level 4 mpid 401 vlan 100`	Provisions a MEP on the interface at maintenance level 4, with an ID of 401, and on VLAN 100.
Step 18	**ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep level 1 mpid 101 vlan 100`	Provisions a MEP on the interface at maintenance level 1 with an ID of 101, and on VLAN 100.
Step 19	**interface** *interface-type interface-id* **Example:** `Router(config-if)# interface gigabitethernet 4/2`	Configures interface Gigabit Ethernet 4/2.
Step 20	**ethernet cfm mip level** *level-id* **Example:** `Router(config)# ethernet cfm mip level 1`	Provisions a MIP on the interface at maintenance level 1.
Step 21	**ethernet cfm cc enable level** {**any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} **Example:** `Router(config)# ethernet cfm cc enable level 4 vlan 100`	Globally enables transmission of CCMs at level 4 on VLAN 100.
Step 22	**ethernet cfm cc enable level** {**any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} **Example:** `Router(config)# ethernet cfm cc enable level 1 vlan 100`	Globally enables transmission of CCMs at level 1 on VLAN 100.

	Command or Action	Purpose
Step 23	**ethernet cfm cc** {**level any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} {**vlan any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num_msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the following parameters for CCMs: • All maintenance levels are to be configured. • All VLANs are to be configured. • The time between CCM transmissions is 20 seconds. • The maximum number of CCMs that can be missed before a MEP is declared down is 3.
Step 24	**Ctrl-Z** **Example:** `Router(config)# Ctrl Z`	Returns the CLI to privileged EXEC mode.

SUMMARY STEPS

PE-AGG A

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

DETAILED STEPS

	Command or Action	Purpose
	PE-AGG A	
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 named OperatorA at level 1 and puts the CLI in Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 65 minutes.
Step 5	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1OpA vlan 100`	Sets a universally unique ID of MetroCustomer1OpA on vlan 100 for a CSI within the maintenance domain OperatorA.
Step 6	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 7	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 8	**interface** *interface-type interface-id* **Example:** `Router(config)# interface gigabitethernet3/1`	Configures interface Gigabit Ethernet 3/1 and enters interface configuration mode.
Step 9	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP at maintenance level 1 on the interface.

	Command or Action	Purpose
Step 10	**interface** *interface-type interface-id* **Example:** `Router(config-if)# interface gigabitethernet4/1`	Configures interface Gigabit Ethernet 4/1.
Step 11	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP at maintenance level 1 on the interface.
Step 12	**Ctrl-Z** **Example:** `Router(config-if)# Ctrl Z`	Returns the CLI to privileged EXEC mode.

SUMMARY STEPS

N-PE A

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

20. **ethernet cfm cc enable level** {**any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*}
21. **ethernet cfm cc** {**level any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} {**vlan any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num_msgs*]
22. **Ctrl-Z**

DETAILED STEPS

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

	Command or Action	Purpose
Step 8	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1OpA vlan 100`	Sets a universally unique ID of MetroCustomer1OpA on VLAN 100 for a CSI within the maintenance domain ServiceProvider.
Step 9	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 10	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 11	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table to 200 messages.
Step 13	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the time that CFM traceroute cache entries are retained to 60 minutes.
Step 14	**interface** *interface-type interface-id* **Example:** `Router(config)# interface gigabitethernet3/0`	Configures interface Gigabit Ethernet 3/0 and puts the CLI in interface configuration mode.
Step 15	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 1`	Provisions a MIP at maintenance level 1 on the interface.
Step 16	**interface** *interface-type interface-id* **Example:** `Router(config-if)# interface gigabitethernet4/0`	Configures interface Gigabit Ethernet 4/0.
Step 17	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 4`	Provisions a MIP at maintenance level 4 on the interface.

	Command or Action	Purpose
Step 18	**ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep level 2 mpid 102 vlan 100`	Sets the interface as a domain boundary (edge) at level 2, defines it as MEP 102, and sets the VLAN to 100.
Step 19	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.
Step 20	**ethernet cfm cc enable level** {**any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} **Example:** `Router(config)# ethernet cfm cc enable level 1 vlan 100`	Globally enables transmission of CCMs at level 1 on VLAN 100.
Step 21	**ethernet cfm cc** {**level any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} {**vlan any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num_msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the following parameters for CCMs: • All maintenance levels are to be configured. • All VLANs are to be configured. • The time between CCM transmissions is 20 seconds. • The maximum number of CCMs that can be missed before a MEP is declared down is 3.
Step 22	**Ctrl-Z** **Example:** `Router(config)# Ctrl Z`	Returns the CLI to privileged EXEC mode.

SUMMARY STEPS

U-PE B

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

11. **ethernet cfm enable**
12. **ethernet cfm traceroute cache**
13. **ethernet cfm traceroute cache size** *entries*
14. **ethernet cfm traceroute cache hold-time** *minutes*
15. **interface** *interface-type interface-id*
16. **ethernet cfm mip level** *level-id*
17. **ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | ,*vlan-id* | *vlan-id-vlan-id* | ,*vlan-id-vlan-id*}
18. **ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | ,*vlan-id* | *vlan-id-vlan-id* | ,*vlan-id-vlan-id*}
19. **exit**
20. **interface** *interface-type interface-id*
21. **ethernet cfm mip level** *level-id*
22. **ethernet cfm cc enable level** {**any** | *level-id* | ,*level-id* | *level-id-level-id* | ,*level-id-level-id*} **vlan** {**any** | *vlan-id* | ,*vlan-id* | *vlan-id-vlan-id* | ,*vlan-id-vlan-id*}
23. **ethernet cfm cc enable level** {**any** | *level-id* | ,*level-id* | *level-id-level-id* | ,*level-id-level-id*} **vlan** {**any** | *vlan-id* | ,*vlan-id* | *vlan-id-vlan-id* | ,*vlan-id-vlan-id*}
24. **ethernet cfm cc** {**level any** | *level-id* | ,*level-id* | *level-id-level-id* | ,*level-id-level-id*} {**vlan any** | *vlan-id* | ,*vlan-id* | *vlan-id-vlan-id* | ,*vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num_msgs*]
25. **Ctrl-Z**

DETAILED STEPS

	Command or Action	Purpose
	U-PE B	
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 named Customer at level 7 and puts the CLI in Ethernet CFM configuration mode.
Step 4	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain ServiceProvider level 4`	Defines a CFM maintenance domain named ServiceProvider at level 4.

	Command or Action	Purpose
Step 5	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 60 minutes.
Step 6	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1 vlan 100`	Sets a universally unique ID of MetroCustomer1 on VLAN 100 for a CSI within the maintenance domain ServiceProvider.
Step 7	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Router(config-ether-cfm)# ethernet cfm domain OperatorB level 2`	Defines a domain named OperatorB at level 2.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 65 minutes.
Step 9	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1OpB vlan 100`	Sets a universally unique ID of MetroCustomer1OpB on VLAN 100 for a CSI within the maintenance domain OperatorB.
Step 10	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 11	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 12	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 13	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table to 200 messages.

	Command or Action	Purpose
Step 14	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the time that CFM traceroute cache entries are retained to 60 minutes.
Step 15	**interface** *interface-type interface-id* **Example:** `Router(config)# interface gigabitethernet1/0`	Configures interface Gigabit Ethernet 1/0 and puts the CLI in interface configuration mode.
Step 16	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 7`	Provisions a MIP at maintenance level 7 on the interface.
Step 17	**ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep level 2 mpid 402 vlan 100`	Sets the interface as a domain boundary (edge) at level 4, defines it as MEP 402, and sets the VLAN to 100.
Step 18	**ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep level 2 mpid 201 vlan 100`	Sets the interface as a domain boundary (edge) at level 2, defines it as MEP 201, and sets the VLAN to 100.
Step 19	**interface** *interface-type interface-id* **Example:** `Router(config-if)# interface gigabitethernet2/0`	Configures interface Gigabit Ethernet 2/0.
Step 20	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at maintenance level 2 on the interface.
Step 21	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.
Step 22	**ethernet cfm cc enable level** {**any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} **Example:** `Router(config)# ethernet cfm cc enable level 4 vlan 100`	Globally enables transmission of CCMs at level 4 on VLAN 100.

	Command or Action	Purpose
Step 23	**ethernet cfm cc enable level** {**any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} **Example:** `Router(config)# ethernet cfm cc enable level 2 vlan 100`	Globally enables transmission of CCMs at level 2 on VLAN 100.
Step 24	**ethernet cfm cc** {**level any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} {**vlan any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num_msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the following parameters for CCMs: • All maintenance levels are to be configured. • All VLANs are to be configured. • The time between CCM transmissions is 20 seconds. • The maximum number of CCMs that can be missed before a MEP is declared down is 3.
Step 25	**Ctrl-Z** **Example:** `Router(config)# Ctrl Z`	Returns the CLI to privileged EXEC mode.

SUMMARY STEPS

PE-AGG B

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

DETAILED STEPS

	Command or Action	Purpose
	PE-AGG B	
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 named OperatorB at level 2 and puts the CLI in Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 65`	Set the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 65 minutes.
Step 5	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1OpB vlan 100`	Sets a universally unique ID of MetroCustomer1OpB on vlan 100 for a CSI within the maintenance domain OperatorB.
Step 6	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 7	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 8	**interface** *interface-type interface-id* **Example:** `Router(config)# interface gigabitethernet1/1`	Configures interface Gigabit Ethernet 1/1 and enters interface configuration mode.
Step 9	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at maintenance level 2 on the interface.

	Command or Action	Purpose
Step 10	**interface** *interface-type interface-id* **Example:** `Router(config-if)# interface gigabitethernet2/1`	Configures interface Gigabit Ethernet 2/1.
Step 11	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at maintenance level 2 on the interface.
Step 12	**Ctrl-Z** **Example:** `Router(config-if)# Ctrl Z`	Returns the CLI to privileged EXEC mode.

SUMMARY STEPS

N-PE B

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

20. **ethernet cfm cc enable level** {**any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*}
21. **ethernet cfm cc** {**level any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} {**vlan any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num_msgs*]
22. **Ctrl-Z**

DETAILED STEPS

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

	Command or Action	Purpose
Step 8	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1OpB vlan 100`	Sets a universally unique ID of MetroCustomer1OpB on vlan 100 for a CSI within the maintenance domain OperatorB.
Step 9	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 10	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 11	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 12	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table to 200 messages.
Step 13	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the time that CFM traceroute cache entries are retained to 60 minutes.
Step 14	**interface** *interface-type interface-id* **Example:** `Router(config)# interface gigabitethernet1/2`	Configures interface Gigabit Ethernet 1/2 and enters interface configuration mode.
Step 15	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 2`	Provisions a MIP at maintenance level 2 on the interface.
Step 16	**interface** *interface-type interface-id* **Example:** `Router(config)# interface gigabitethernet2/2`	Configures interface Gigabit Ethernet 2/2 and enters interface configuration mode.
Step 17	**ethernet cfm mip level** *level-id* **Example:** `Router(config-if)# ethernet cfm mip level 4`	Provisions a MIP at maintenance level 4 on the interface.

	Command or Action	Purpose
Step 18	**ethernet cfm mep level** *level-id* [**inward**] **mpid** *id* **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} **Example:** `Router(config-if)# ethernet cfm mep level 2 mpid 202 vlan 100`	Sets the interface as a domain boundary (edge), defines it as MEP 202, and sets the VLAN to 100.
Step 19	**exit** **Example:** `Router(config-if)# exit`	Returns the CLI to global configuration mode.
Step 20	**ethernet cfm cc enable level** {**any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} **vlan** {**any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} **Example:** `Router(config)# ethernet cfm cc enable level 2 vlan 100`	Globally enables transmission of CCMs at level 2 on VLAN 100.
Step 21	**ethernet cfm cc** {**level any** | *level-id* | *,level-id* | *level-id-level-id* | *,level-id-level-id*} {**vlan any** | *vlan-id* | *,vlan-id* | *vlan-id-vlan-id* | *,vlan-id-vlan-id*} [**interval** *seconds*] [**loss-threshold** *num_msgs*] **Example:** `Router(config)# ethernet cfm cc level any vlan any interval 20 loss-threshold 3`	Sets the following parameters for CCMs: • All maintenance levels are to be configured. • All VLANs are to be configured. • The time between CCM transmissions is 20 seconds. • The maximum number of CCMs that can be missed before a MEP is declared down is 3.
Step 22	**Ctrl-Z** **Example:** `Router(config)# Ctrl Z`	Returns the CLI to privileged EXEC mode.

SUMMARY STEPS

CE-B

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**]
4. **mep archive-hold-time** *minutes*
5. **service** *csi-id* **vlan** *vlan-id*
6. **exit**
7. **ethernet cfm enable**
8. **ethernet cfm traceroute cache**
9. **ethernet cfm traceroute cache size** *entries*
10. **ethernet cfm traceroute cache hold-time** *minutes*

11. **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*}
12. **Ctrl-Z**

DETAILED STEPS

	Command or Action	Purpose
	CE-B	
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain Customer level 7 direction outward`	Defines an outward CFM maintenance domain named Customer at level 7 and puts the CLI in configuration Ethernet CFM configuration mode.
Step 4	**mep archive-hold-time** *minutes* **Example:** `Router(config-ether-cfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged to 60 minutes.
Step 5	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service MetroCustomer1 vlan 100`	Sets a universally unique ID of MetroCustomer1 on VLAN 100 for a CSI within the maintenance domain ServiceProvider.
Step 6	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 7	**ethernet cfm enable** **Example:** `Router(config)# ethernet cfm enable`	Enables CFM processing globally on the device.
Step 8	**ethernet cfm traceroute cache** **Example:** `Router(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.

	Command or Action	Purpose
Step 9	**ethernet cfm traceroute cache size** *entries* **Example:** `Router(config)# ethernet cfm traceroute cache size 200`	Sets the maximum size for the CFM traceroute cache table to 200 messages.
Step 10	**ethernet cfm traceroute cache hold-time** *minutes* **Example:** `Router(config)# ethernet cfm traceroute cache hold-time 60`	Sets the time that CFM traceroute cache entries are retained to 60 minutes and places the CLI in interface configuration mode.
Step 11	**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 702 vlan 100`	Provisions a MEP on the interface at maintenance level 7, in outward domain Customer, with an ID of 702, and on VLAN 100.
Step 12	**Ctrl-Z** **Example:** `Router(config)# Ctrl Z`	Returns the CLI to privileged EXEC mode.

Configuring and Enabling the Cross-Check Function

Perform this task to configure and enable cross-checking for an inward facing 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 Inward Facing MEP

SUMMARY STEPS

U-PE A

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. **end**
6. **ethernet cfm mep crosscheck start-delay** *delay*
7. **Ctrl Z**
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
	U-PE A	
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 named ServiceProvider at level 4 and puts the CLI in Ethernet CFM configuration mode.
Step 4	**mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*] **Example:** `Router(config-ether-cfm)# mep crosscheck mpid 402 vlan 100`	Statically defines a remote MEP with an ID of 402 on VLAN 100 within the ServiceProvider domain.
Step 5	**end** **Example:** `Router(config-ether-cfm)# end`	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 at 60 seconds.
Step 7	**Ctrl Z** **Example:** `Router(config)# Ctrl Z`	Returns the CLI to privileged EXEC mode.
Step 8	**ethernet cfm mep crosscheck** {**enable** | **disable**} **level** {*level-id* | *level-id-level-id* [,*level-id-level-id*]} **vlan** {*vlan-id* | **any** | *vlan-id-vlan-id* [,*vlan-id-vlan-id*]} **Example:** `Router# ethernet cfm mep crosscheck enable level 4 vlan 100`	Enables cross-checking between remote MEP 402 in the ServiceProvider domain and MEPs learned through CCMs on VLAN 100.

SUMMARY STEPS

U-PE B

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. **end**
6. **ethernet cfm mep crosscheck start-delay** *delay*
7. **Ctrl Z**
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
	U-PE B	
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 named ServiceProvider at level 4 and puts the CLI in Ethernet CFM configuration mode.
Step 4	**mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*] **Example:** `Router(config-ether-cfm)# mep crosscheck mpid 401 vlan 100`	Statically defines a remote MEP with an ID of 401 on VLAN 100 within the ServiceProvider domain.
Step 5	**end** **Example:** `Router(config-ether-cfm)# end`	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 at 60 seconds.

	Command or Action	Purpose
Step 7	**Ctrl Z** **Example:** `Router(config)# Ctrl Z`	Returns the CLI to privileged EXEC mode.
Step 8	**ethernet cfm mep crosscheck** {**enable** | **disable**} **level** {*level-id* | *level-id-level-id* [,*level-id-level-id*]} **vlan** {*vlan-id* | **any** | *vlan-id-vlan-id* [,*vlan-id-vlan-id*]} **Example:** `Router# ethernet cfm mep crosscheck enable level 4 vlan 100`	Enables cross-checking between remote MEP 401 in the ServiceProvider domain and MEPs learned through CCMs on VLAN 100.

Configuring and Enabling Cross-Checking for an Outward Facing MEP

SUMMARY STEPS

CE-A

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. **end**
6. **ethernet cfm mep crosscheck start-delay** *delay*
7. **Ctrl Z**
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
	CE-A	
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain Customer level 7 direction outward`	Defines an outward CFM domain named Customer at level 7 and puts the CLI in Ethernet CFM configuration mode.
Step 4	**mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*] **Example:** `Router(config-ether-cfm)# mep crosscheck mpid 702 vlan 100`	Statically defines a remote MEP with an ID of 702 on VLAN 100 within the Customer domain.
Step 5	**end** **Example:** `Router(config-ether-cfm)# end`	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 at 60 seconds.
Step 7	**Ctrl Z** **Example:** `Router(config)# Ctrl Z`	Returns the CLI to privileged EXEC mode.
Step 8	**ethernet cfm mep crosscheck** {**enable** | **disable**} **level** {*level-id* | *level-id-level-id* [,*level-id-level-id*]} **vlan** {*vlan-id* | **any** | *vlan-id-vlan-id* [,*vlan-id-vlan-id*]} **Example:** `Router# ethernet cfm mep crosscheck enable level 7 vlan 100`	Enables cross-checking between remote MEP 702 in the Customer domain and MEPs learned through CCMs on VLAN 100.

SUMMARY STEPS

CE-B

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. **end**
6. **ethernet cfm mep crosscheck start-delay** *delay*

7. **Ctrl Z**
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
	CE-B	
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* [**direction outward**] **Example:** `Router(config)# ethernet cfm domain Customer level 7 direction outward`	Defines an outward CFM domain named Customer at level 7 and puts the CLI in Ethernet CFM configuration mode.
Step 4	**mep crosscheck mpid** *id* **vlan** *vlan-id* [**mac** *mac-address*] **Example:** `Router(config-ether-cfm)# mep crosscheck mpid 401 vlan 100`	Statically defines a remote MEP with an ID of 401 on VLAN 100 within the ServiceProvider domain.
Step 5	**end** **Example:** `Router(config-ether-cfm)# end`	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 at 60 seconds.

	Command or Action	Purpose
Step 7	**Ctrl Z** **Example:** `Router(config)# Ctrl Z`	Returns the CLI to privileged EXEC mode.
Step 8	**ethernet cfm mep crosscheck** {**enable** | **disable**} **level** {*level-id* | *level-id-level-id* [,*level-id-level-id*]} **vlan** {*vlan-id* | **any** | *vlan-id-vlan-id* [,*vlan-id-vlan-id*]} **Example:** `Router# ethernet cfm mep crosscheck enable level 7 vlan 100`	Enables cross-checking between remote MEP 401 in the Customer domain and MEPs learned through CCMs on VLAN 100.

Examples

Configuring Cross-Checking on an Inward Facing MEP

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

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

Enabling Cross-Checking on an Inward Facing MEP

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

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

Troubleshooting Tips

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

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

Configuring Ethernet OAM Interaction with CFM

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

Configuring the OAM Manager

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

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

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
Step 4	**service** *csi-id* **vlan** *vlan-id* **Example:** `Router(config-ether-cfm)# service csi2 vlan 10`	Defines a universally unique customer service instance (CSI) and VLAN ID within the maintenance domain.
Step 5	**exit** **Example:** `Router(config-ether-cfm)# exit`	Returns the CLI to global configuration mode.
Step 6	**ethernet evc** *evc-id* **Example:** `Router(config)# ethernet evc 50`	Defines an EVC and places the CLI in EVC configuration mode.
Step 7	**oam protocol** {**cfm svlan** *svlan-id* **domain** *domain-name* | **ldp**} **Example:** `Router(config-evc)# oam protocol cfm svlan 10 domain cstmr1`	Configures the EVC OAM protocol.
Step 8	**exit** **Example:** `Router(config-evc)# exit`	Returns the CLI to global configuration mode.
Step 9	Repeat Steps 3 through 8 to define other CFM domains that you want OAM manager to monitor.	
Step 10	**end** **Example:** `Router(config)# end`	Returns the CLI to privileged EXEC mode.

Enabling Ethernet OAM

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

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

SUMMARY STEPS

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

DETAILED STEPS

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

Configuration Examples for Ethernet Connectivity Fault Management

The following are examples of provisioning a network and provisioning service.

- Provisioning a Network: Example, page 65
- Provisioning Service: Example, page 68

Provisioning a Network: Example

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

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

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

U-PE A

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

PE-AGG A

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

N-PE A

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

U-PE B

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

PE-AGG B

```
ethernet cfm domain OperatorB level 2
mep archive-hold-time 65
!
ethernet cfm enable
!
interface gigabitethernet1/1
ethernet cfm mip level 2
!
interface gigabitethernet2/1
ethernet cfm mip level 2
```

N-PE B

```
!
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
!
ethernet cfm domain ServiceProvider level 4
mep archive-hold-time 60
!
ethernet cfm domain OperatorB level 2
mep archive-hold-time 65
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
interface gigabitethernet1/2
ethernet cfm mip level 2
!
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up
```

CE-B

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

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

Provisioning Service: Example

CE-A

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

U-PE A

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

PE-AGG A

```
ethernet cfm domain OperatorA level 1
mep archive-hold-time 65
service MetroCustomer1OpA vlan 100
!
ethernet cfm enable
!
interface gigabitethernet3/1
ethernet cfm mip level 1
```

```
!
interface gigabitethernet4/1
ethernet cfm mip level 1
```

N-PE A

```
!
ethernet cfm domain ServiceProvider level 4
mep archive-hold-time 60
service MetroCustomer1 vlan 100
!
ethernet cfm domain OperatorA level 1
mep archive-hold-time 65
service MetroCustomer1OpA vlan 100
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
interface gigabitethernet3/0
ethernet cfm mip level 1
!
interface gigabitethernet4/0
ethernet cfm mip level 4
ethernet cfm mep level 1 mpid 102 vlan 100
!
ethernet cfm cc enable level 1 vlan 100
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
```

U-PE B

```
!
ethernet cfm domain Customer level 7
!
ethernet cfm domain ServiceProvider level 4
mep archive-hold-time 60
service MetroCustomer1 vlan 100
!
ethernet cfm domain OperatorB level 2
mep archive-hold-time 65
service MetroCustomer1OpB vlan 100
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
interface gigabitethernet1/0
ethernet cfm mip level 7
ethernet cfm mep level 4 mpid 402 vlan 100
ethernet cfm mep level 2 mpid 201 vlan 100
!
interface gigabitethernet2/0
ethernet cfm mip level 2
!
ethernet cfm cc enable level 4 vlan 100
ethernet cfm cc enable level 2 vlan 100
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
```

PE-AGG B

```
ethernet cfm domain OperatorB level 2
mep archive-hold-time 65
service MetroCustomer1OpB vlan 100
!
ethernet cfm enable
```

```
!
interface gigabitethernet1/1
ethernet cfm mip level 2
!
interface gigabitethernet2/1
ethernet cfm mip level 2
```

N-PE B

```
!
ethernet cfm domain ServiceProvider level 4
mep archive-hold-time 60
service MetroCustomer1 vlan 100
!
ethernet cfm domain OperatorB level 2
mep archive-hold-time 65
service MetroCustomer1OpB vlan 100
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
interface gigabitethernet1/2
ethernet cfm mip level 2
!
interface gigabitethernet2/2
ethernet cfm mip level 4
ethernet cfm mep level 2 mpid 202 vlan 100
!
ethernet cfm cc enable level 2 vlan 100
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
```

CE-B

```
!
ethernet cfm domain Customer level 7 direction outward
service Customer1 vlan 100
!
ethernet cfm enable
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!
interface gigabitethernet3/2
ethernet cfm mep level 7 direction outward domain Customer1 mpid 702 vlan 100
!
ethernet cfm cc enable level 7 vlan 100
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
```

Additional References

The following sections provide references related to Ethernet Connectivity Fault Management.

Related Documents

Related Topic	Document Title
IEEE 802.3ah	*IEEE 802.3ah Ethernet in the First Mile*
CFM commands	*Cisco IOS Carrier Ethernet Command Reference*, Release 12.4T
Ethernet local management interface on a provider edge device.	*Configuring Ethernet Local Management Interface on a Provider Edge Device*

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*
ITU-T	*ITU-T Y.1731 OAM Mechanisms for Ethernet-Based Networks*
IETF VPLS OAM	*L2VPN OAM Requirements and Framework*

MIBs

MIB	MIBs Link
CISCO-ETHER-CFM-MIB	To locate and download MIBs for selected platforms, Cisco IOS 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 website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	http://www.cisco.com/techsupport

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/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 a bridge portion an operator’s bridge that is connected to 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.

Note See *Internetworking Terms and Acronyms* for terms not included in this glossary.

Feature Information for Ethernet Connectivity Fault Management

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(33)SRA, Cisco IOS Release 12.4(11)T, Cisco IOS Release 12.2(33)SRB or a later release appear in the table.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Ethernet Connectivity Fault Management

Feature Name	Releases	Feature Information
Ethernet Connectivity Fault Management	12.2(33)SRA	Ethernet CFM is an end-to-end per-service-instance Ethernet layer OAM protocol. It includes proactive connectivity monitoring, fault verification, and fault isolation for large Ethernet MANs and WANs. Ethernet CFM is supported on the Cisco 7600 router in Cisco IOS Release 12.2(33)SRA. The following sections provide information about this feature: • Ethernet CFM, page 3 • Customer Service Instance, page 4 • Maintenance Domain, page 4 • Maintenance Point, page 6 • CFM Messages, page 8 • Cross-Check Function, page 9 • SNMP Traps, page 9 • Designing CFM Domains, page 11 • Configuring Ethernet CFM, page 13

Table 1 Feature Information for Ethernet Connectivity Fault Management (continued)

Feature Name	Releases	Feature Information
Outward Facing MEP	12.4(11)T 12.2(33)SRB	The Outward Facing MEP feature is an enhancement to Ethernet CFM that supports the distribution and access environments by supporting outward facing MEPs on routed (Layer 3) ports. Outward facing MEPs can also be used to monitor network-to-network interfaces between operators on Layer 2 switch ports. Ethernet CFM with support for outward facing MEPs is supported on the Cisco Integrated Services Routers (ISRs) in Cisco IOS Release 12.4(11)T. The following sections provide information about this feature: • Maintenance Domain, page 4 • Maintenance Endpoints, page 6 • Outward Facing MEPs, page 7 • Provisioning the Network, page 13 (CE-A and CE-B) • Provisioning Service, page 34 (CE-A and CE-B) • Configuring and Enabling Cross-Checking for an Outward Facing MEP, page 59
Ethernet OAM and Ethernet CFM Interworking	12.2(33)SRB	The Ethernet OAM and Ethernet CFM Interworking feature enables Ethernet OAM and CFM to function together in a network. Support for this feature was introduced in Cisco IOS Release 12.2(33)SRB.