



Configuring Ethernet Connectivity Fault Management in a Service Provider Network

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

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



Note

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

- [Prerequisites for Configuring Ethernet CFM in a Service Provider Network, page 2](#)
- [Restrictions for Configuring Ethernet CFM in a Service Provider Network, page 2](#)
- [Information About Configuring Ethernet CFM in a Service Provider Network, page 3](#)
- [How to Set Up Ethernet CFM in a Service Provider Network, page 12](#)
- [Configuration Examples for Configuring Ethernet CFM in a Service Provider Network, page 85](#)
- [Glossary, page 90](#)

Prerequisites for Configuring Ethernet CFM in a Service Provider Network

Business Requirements

- Network topology and network administration have been evaluated.
- Business and service policies have been established.
- Partial Route Computation (PRC) codes have been implemented for all supported commands related to configuring High Availability (HA) on a maintenance endpoint (MEP), maintenance intermediate point (MIP), level, service instance ID, cross-check timer, cross-check, and domain.

Restrictions for Configuring Ethernet CFM in a Service Provider Network

- CFM loopback messages will not be confined within a maintenance domain according to their maintenance level. The impact of not having CFM loopback messages confined to their maintenance levels occurs at these levels:
 - Architecture—CFM layering is violated for loopback messages.
 - Deployment—A user may potentially misconfigure a network and have loopback messages succeed.
 - Security—A malicious device that recognizes devices' MAC addresses and levels may potentially explore a network topology that should be transparent.
- CFM is not fully supported on a Multiprotocol Label Switching (MPLS) provider edge (PE) device. There is no interaction between CFM and an Ethernet over MPLS (EoMPLS) pseudowire.
- CFM configuration is not supported on an EtherChannel in FastEthernet Channel (FEC) mode.
- The HA features NFS/SSO Support in CFM 802.1ag/1.0d and ISSU Support in CFM 802.1ag/1.0d are not supported on customer edge (CE) devices.
- The NFS/SSO Support in CFM 802.1ag/1.0d feature is not supported for the traceroute and error databases.

Information About Configuring Ethernet CFM in a Service Provider Network

Ethernet CFM

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

Being an end-to-end technology is the distinction between CFM and other metro-Ethernet OAM protocols. For example, MPLS, ATM, and SONET OAM help in debugging Ethernet wires but are not always end-to-end. 802.3ah OAM is a single-hop and per-physical-wire protocol. It is not end to end or service aware.

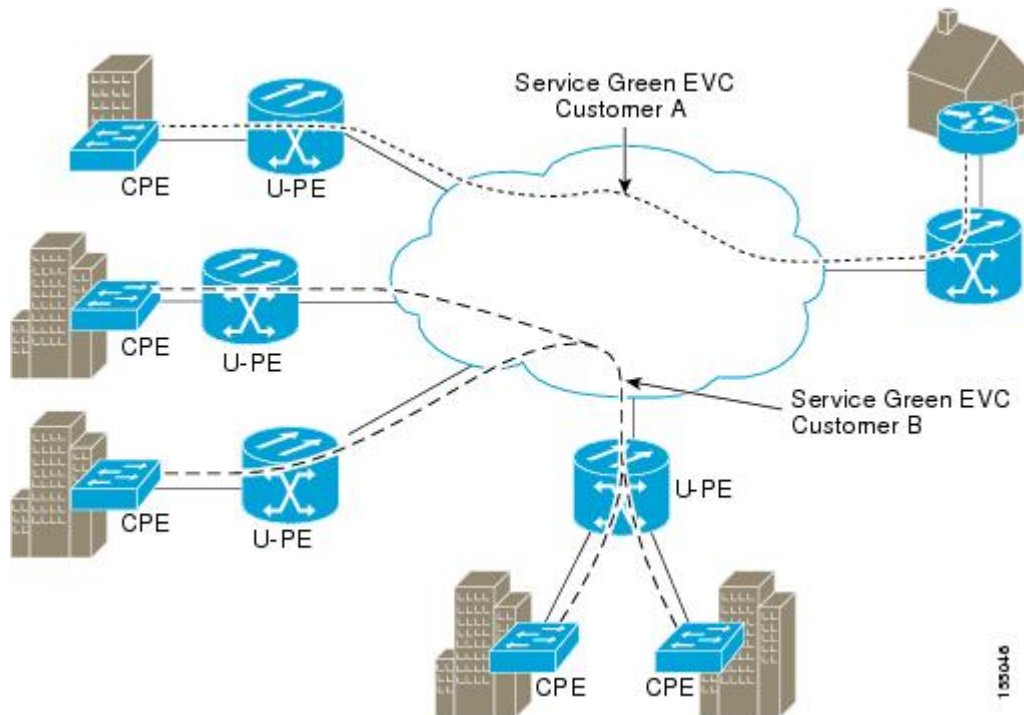
Troubleshooting carrier networks offering Ethernet Layer 2 services is challenging. Customers contract with service providers for end-to-end Ethernet service and service providers may subcontract with operators to provide equipment and networks. Compared to enterprise networks, where Ethernet traditionally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

Benefits of Ethernet CFM

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

Customer Service Instance

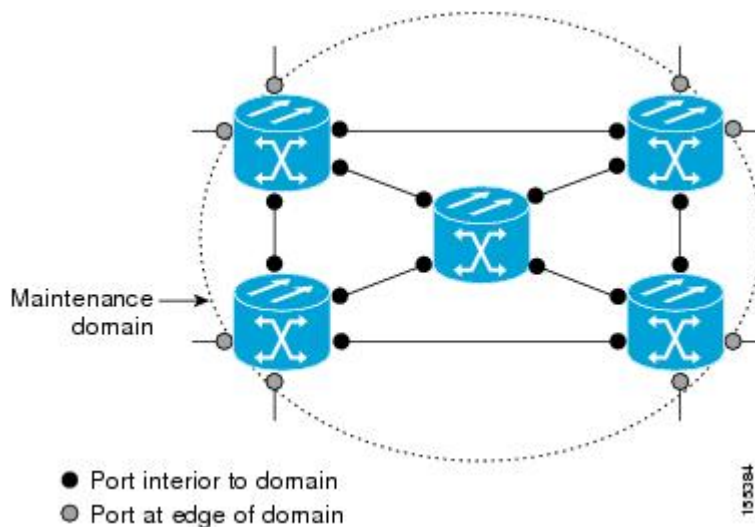
A customer service instance is an Ethernet virtual connection (EVC), which is identified by an S-VLAN within an Ethernet island, and is identified by a globally unique service ID. A customer service instance can be point-to-point or multipoint-to-multipoint. The figure below shows two customer service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



135046

Maintenance Domain

A maintenance domain is a management space for the purpose of managing and administering a network. A domain is owned and operated by a single entity and defined by the set of ports internal to it and at its boundary. The figure below illustrates a typical maintenance domain.



135384

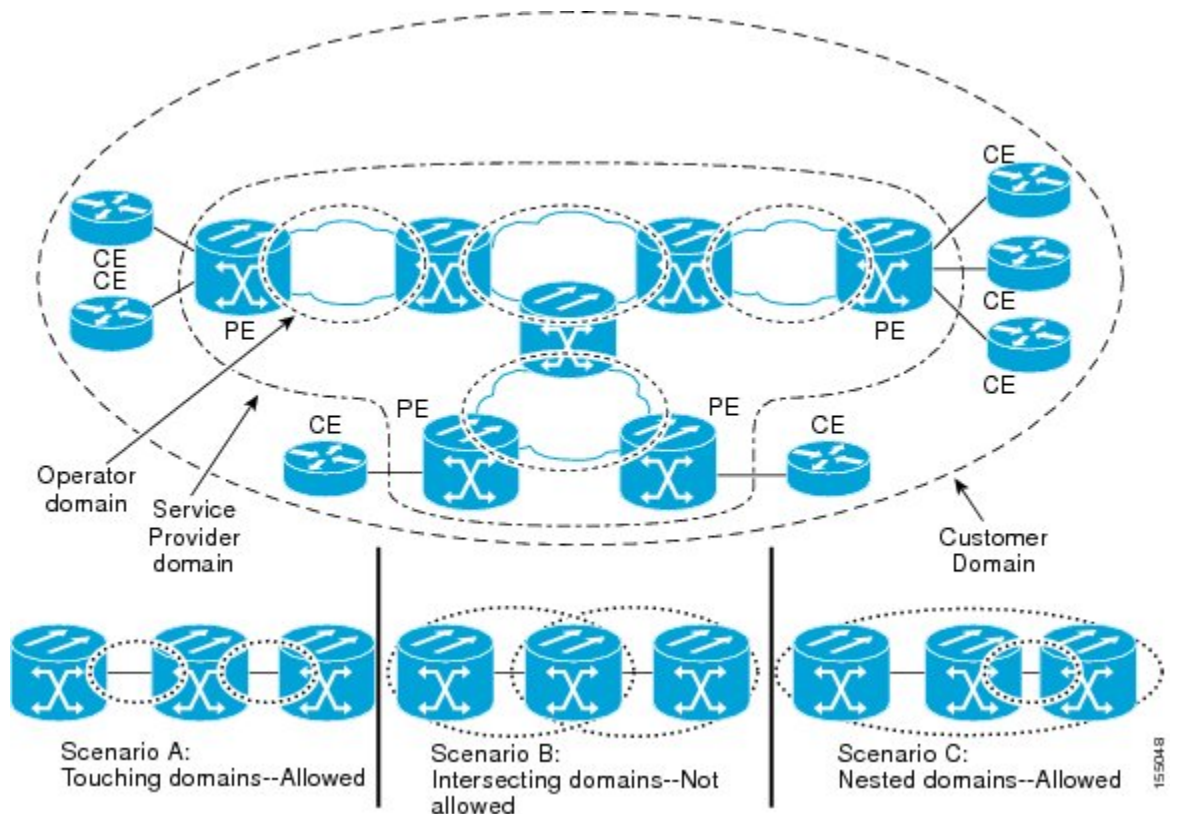
A unique maintenance level in the range of 0 to 7 is assigned to each domain by a network administrator. Levels and domain names are useful for defining the hierarchical relationship that exists among domains. The hierarchical relationship of domains parallels the structure of customer, service provider,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

larger the domain, the higher the level value. For example, a customer domain would be larger than an operator domain. The customer domain may have a maintenance level of 7 and the operator domain may have a maintenance level of 0. Typically, operators would have the smallest domains and customers the largest domains, with service provider domains between them in size. All levels of the hierarchy must operate together.

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

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

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



Maintenance Point

A maintenance point is a demarcation point on an interface (port) that participates in CFM within a maintenance domain. Maintenance points on device ports act as filters that confine CFM frames within the bounds of a

domain by dropping frames that do not belong to the correct level. Maintenance points must be explicitly configured on Cisco devices. Two classes of maintenance points exist, MEPs and MIPs.

Maintenance Endpoints

Maintenance endpoints (MEPs) have the following characteristics:

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

Inward Facing MEPs

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

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



Note

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

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

Outward Facing MEPs for Port Channels

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

Outward facing MEPs on port channels use the Bridge-Brain MAC address of the first member link. When port channel members change, the identities of outward facing MEPs do not have to change.

An outward facing MEP performs the following functions:

- Sends and receives CFM frames at its level via the wire connected to the port where the MEP is configured.

- Drops all CFM frames at its level (or at a lower level) that come from the direction of the relay function.
- Processes all CFM frames at its level coming from the direction of the wire.
- Drops all CFM frames at a lower level coming from the direction of the wire.
- Transparently forwards all CFM frames at levels higher than the level of the outward facing MEP, independent of whether they come in from the relay function side or the wire side.
- If the port on which the outward MEP is configured is blocked by the Spanning-Tree Protocol, the MEP can still transmit and receive CFM messages via the wire.

Maintenance Intermediate Points

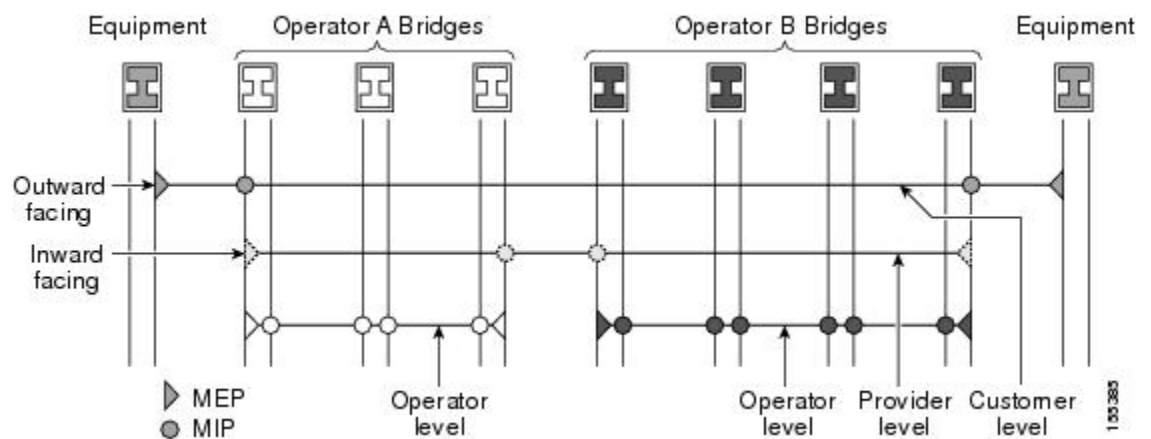
MIPs have the following characteristics:

- Per maintenance domain (level) and for all S-VLANs enabled or allowed on a port.
- Internal to a domain, not at the boundary.
- CFM frames received from MEPs and other MIPs are cataloged and forwarded, using both the wire and the relay function.
- All CFM frames at a lower level are stopped and dropped, independent of whether they originate from the wire or relay function.
- All CFM frames at a higher level are forwarded, independent of whether they arrive from the wire or relay function.
- Passive points respond only when triggered by CFM traceroute and loopback messages.
- Bridge-Brain MAC addresses are used.

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

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

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



CFM Messages

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

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

- Continuity Check
- Loopback
- Traceroute

Continuity Check Messages

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

CFM CCMs have the following characteristics:

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

Loopback Messages

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

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

Traceroute Messages

CFM traceroute messages are multicast frames that a MEP transmits, at the request of an administrator, to track the path (hop-by-hop) to a destination MEP. They allow the transmitting node to discover vital connectivity data about the path, and allow the discovery of all MIPs along the path that belong to the same maintenance

domain. For each visible MIP, traceroute messages indicate ingress action, relay action, and egress action. Traceroute messages are similar in concept to User Datagram Protocol (UDP) traceroute messages.

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

Cross-Check Function

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

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

SNMP Traps

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

CC Traps

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

Cross-Check Traps

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

Ethernet CFM and Ethernet OAM Interaction

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

Ethernet Virtual Circuit

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

OAM Manager

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

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

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

CFM over Bridge Domains

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

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

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

HA Features Supported by CFM

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



Note

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

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

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

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

Benefits of CFM HA

- Elimination of network downtime for Cisco software image upgrades, allowing for faster upgrades.
- Elimination of resource scheduling challenges associated with planned outages and late night maintenance windows.
- Accelerated deployment of new services and applications and facilitation of faster implementation of new features.
- Reduced operating costs due to outages while delivering higher service levels.
- CFM updates its databases and controls its own HA messaging and versioning, and this control facilitates maintenance.

CFM HA in a Metro Ethernet Network

A standalone Connectivity Fault Management (CFM) implementation does not have explicit high availability (HA) requirements. When CFM is implemented on a customer edge (CE) or provider edge (PE), CFM must maintain the Ethernet virtual circuit (EVC) state, which requires HA because the EVC state is critical in maintaining end-to-end connectivity. CFM configures the platform with maintenance level, domain, and maintenance point, learns the remote maintenance point information, and maps it to the appropriate EVC. CFM then aggregates data received from all remote ports; consequently HA requirements vary for CE and PE.

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

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

**Note**

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

NSF SSO Support in CFM 802.1ag 1.0d

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

networking device, or is manually taken down for maintenance. NSF interoperates with the SSO feature to minimize network downtime following a switchover. The primary function of Cisco NSF is to continue forwarding IP packets following an RSP switchover.

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

ISSU Support in CFM 802.1ag 1.0d

In Service Upgrades (ISSUs) allow you to perform a Cisco software upgrade or downgrade without disrupting packet flow. Connectivity Fault Management (CFM) performs a bulk update and a runtime update of the continuity check database to the standby Route Switch Processor (RSP), including adding, deleting, or updating a row. This checkpoint data requires ISSU capability to transform messages from one release to another. All the components that perform active RSP to standby RSP updates using messages require ISSU support.

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

How to Set Up Ethernet CFM in a Service Provider Network

Designing CFM Domains

**Note**

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

Before You Begin

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

SUMMARY STEPS

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

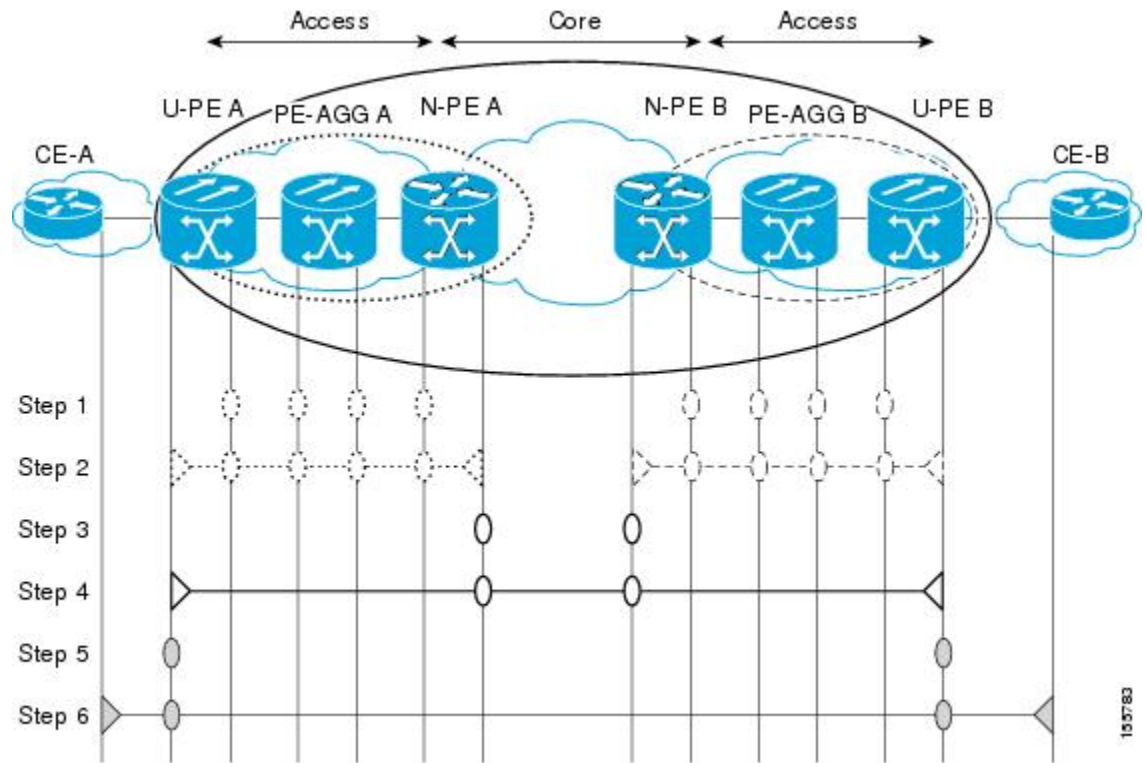
DETAILED STEPS

	Command or Action	Purpose
Step 1	Determine operator level MIPs.	<p>Follow these steps:</p> <ul style="list-style-type: none"> • Starting at lowest operator level domain, assign a MIP at every interface internal to the operator network to be visible to CFM. • Proceed to next higher operator level and assign MIPs. • Verify that every port that has a MIP at a lower level does not have maintenance points at a higher level. • Repeat steps a through d until all operator MIPs are determined.
Step 2	Determine operator level MEPs.	<p>Follow these steps:</p> <ul style="list-style-type: none"> • Starting at the lowest operator level domain, assign a MEP at every UNI that is part of a service instance. • Assign a MEP at the network to network interface (NNI) between operators, if there is more than one operator. • Proceed to next higher operator level and assign MEPs. • A port with a MIP at a lower level cannot have maintenance points at a higher level. A port with a MEP at a lower level should have either a MIP or MEP at a higher level.
Step 3	Determine service provider MIPs.	<p>Follow these steps:</p> <ul style="list-style-type: none"> • Starting at the lowest service provider level domain, assign service provider MIPs at the NNI between operators (if more than one). • Proceed to next higher service provider level and assign MIPs. • A port with a MIP at a lower level cannot have maintenance points at a higher level. A port with a MEP at a lower level should not have either a MIP or a MEP at a higher level.
Step 4	Determine service provider MEPs.	<p>Follow these steps:</p> <ul style="list-style-type: none"> • Starting at the lowest service provider level domain, assign a MEP at every UNI that is part of a service instance.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • 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.	<p>Customer MIPs are allowed only on the UNIs at the uPEs if the service provider allows the customer to run CFM. Otherwise, the service provider can configure Cisco devices to block CFM frames.</p> <ul style="list-style-type: none"> • Configure a MIP on every uPE, at the UNI port, in the customer maintenance domain. • Ensure the MIPs are at a maintenance level that is at least one higher than the highest level service provider domain.
Step 6	Determine customer MEPs.	Customer MEPs are on customer equipment. Assign an outward facing MEP within an outward domain at the appropriate customer level at the handoff between the service provider and the customer.

Examples

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



What to Do Next

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

Configuring Ethernet CFM

Configuring Ethernet CFM consists of the following tasks:

Provisioning the Network

Provisioning the Network on the CE-A

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
Step 4	<p>service {<i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i>} {vlan <i>vlan-id</i> port evc <i>evc-name</i>} direction {up down}</p> <p>Example: Device(config-ecfm)# service s41 evc 41 vlan 41</p>	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	<p>continuity-check</p> <p>Example: Device(config-ecfm-srv)# continuity-check</p>	Configures the transmission of continuity check messages (CCMs).
Step 6	<p>continuity-check [interval <i>cc-interval</i>]</p> <p>Example: Device(config-ecfm-srv)# continuity-check interval 10s</p>	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	<p>exit</p> <p>Example: Device(config-ecfm-srv)# exit</p>	Returns to Ethernet connectivity fault management configuration mode.
Step 8	<p>mep archive-hold-time <i>minutes</i></p> <p>Example: Device(config-ecfm)# mep archive-hold-time 60</p>	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	<p>exit</p> <p>Example: Device(config-ecfm)# exit</p>	Returns to global configuration mode.
Step 10	<p>ethernet cfm global</p> <p>Example: Device(config)# ethernet cfm global</p>	Enables CFM processing globally on the device.
Step 11	<p>ethernet cfm traceroute cache</p> <p>Example: Device(config)# ethernet cfm traceroute cache</p>	Enables caching of CFM data learned through traceroute messages.

	Command or Action	Purpose
Step 12	<p>ethernet cfm traceroute cache size <i>entries</i></p> <p>Example:</p> <pre>Device(config)# ethernet cfm traceroute cache size 200</pre>	Sets the maximum size for the CFM traceroute cache table.
Step 13	<p>ethernet cfm traceroute cache hold-time <i>minutes</i></p> <p>Example:</p> <pre>Device(config)# ethernet cfm traceroute cache hold-time 60</pre>	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	<p>snmp-server enable traps ethernet cfm cc [mep-up] [mep-down] [config] [loop] [cross-connect]</p> <p>Example:</p> <pre>Device(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect</pre>	Enables SNMP trap generation for Ethernet CFM continuity check events.
Step 15	<p>snmp-server enable traps ethernet cfm crosscheck [mep-unknown mep-missing service-up]</p> <p>Example:</p> <pre>Device(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up</pre>	Enables SNMP trap generation for Ethernet CFM continuity check events in relation to the cross-check operation between statically configured MEPS and those learned via CCMs.
Step 16	<p>end</p> <p>Example:</p> <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.

Provisioning the Network on the U-PE A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **service** {*short-ma-name* | **number** *MA-number* | **vlan-id** *primary-vlan-id* | **vpn-id** *vpn-id*} {**vlan** *vlan-id* | **port** | **evc** *evc-name*} **direction** {**up** | **down**}
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
11. **ethernet cfm traceroute cache**
12. **ethernet cfm traceroute cache** **size** *entries*
13. **ethernet cfm traceroute cache** **hold-time** *minutes*
14. **interface** *type number*
15. **service instance** *id* **ethernet** [*evc-name*]
16. **encapsulation** *encapsulation-type*
17. **bridge-domain** *bridge-id*
18. **cfm mip level** { *level* }
19. **exit**
20. **exit**
21. **snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**]
22. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown** | **mep-missing** | **service-up**]
23. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: <pre>Device(config)# ethernet cfm domain Customer level 7</pre>	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	service { <i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i> } { vlan <i>vlan-id</i> port evc <i>evc-name</i> } direction { up down } Example: <pre>Device(config-ecfm)# service s41 evc 41 vlan 41</pre>	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	continuity-check Example: <pre>Device(config-ecfm-srv)# continuity-check</pre>	Configures the transmission of continuity check messages (CCMs).
Step 6	continuity-check [interval <i>cc-interval</i>] Example: <pre>Device(config-ecfm-srv)# continuity-check interval 10s</pre>	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	exit Example: <pre>Device(config-ecfm-srv)# exit</pre>	Returns to Ethernet connectivity fault management configuration mode.
Step 8	mep archive-hold-time <i>minutes</i> Example: <pre>Device(config-ecfm)# mep archive-hold-time 60</pre>	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	exit Example: <pre>Device(config-ecfm)# exit</pre>	Returns to global configuration mode.
Step 10	ethernet cfm global Example: <pre>Device(config)# ethernet cfm global</pre>	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 11	ethernet cfm traceroute cache Example: Device(config)# ethernet cfm traceroute cache	Enables caching of CFM data learned through traceroute messages.
Step 12	ethernet cfm traceroute cache size entries Example: Device(config)# ethernet cfm traceroute cache size 200	Sets the maximum size for the CFM traceroute cache table.
Step 13	ethernet cfm traceroute cache hold-time minutes Example: Device(config)# ethernet cfm traceroute cache hold-time 60	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	interface type number Example: Device(config)# interface gigabitethernet4/2	Specifies an interface and enters interface configuration mode.
Step 15	service instance id ethernet [evc-name] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	encapsulation encapsulation-type Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 17	bridge-domain bridge-id Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 18	cfm mip level { level } Example: Device(config-if-srv)# cfm mip level 4	Creates a MIP and sets the maintenance level number.

	Command or Action	Purpose
Step 19	exit Example: Device(config-if-srv)# exit	Returns to interface configuration mode.
Step 20	exit Example: Device(config-if)# exit	Returns to global configuration mode.
Step 21	snmp-server enable traps ethernet cfm cc [mep-up] [mep-down] [config] [loop] [cross-connect] Example: Device(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.
Step 22	snmp-server enable traps ethernet cfm crosscheck [mep-unknown mep-missing service-up] Example: Device(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.
Step 23	end Example: Device(config)# end	Returns to privileged EXEC mode.

Provisioning the Network on the PE-AGG A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **service** {*short-ma-name* | **number** *MA-number* | **vlan-id** *primary-vlan-id* | **vpn-id** *vpn-id*} {**vlan** *vlan-id* | **port** | **evc** *evc-name*} **direction** {**up** | **down**}
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
11. **interface** *type* *number*
12. **service instance** *id* **ethernet** [*evc-name*]
13. **encapsulation** *encapsulation-type*
14. **bridge-domain** *bridge-id*
15. **cfm mip level** *level*
16. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: Device(config)# ethernet cfm domain Customer level 7	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 4	service { <i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i> } { vlan <i>vlan-id</i> port evc <i>evc-name</i> } direction { up down } Example: Device(config-ecfm)# service s41 evc 41 vlan 41	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	continuity-check Example: Device(config-ecfm-srv)# continuity-check	Configures the transmission of continuity check messages (CCMs).
Step 6	continuity-check [interval <i>cc-interval</i>] Example: Device(config-ecfm-srv)# continuity-check interval 10s	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	exit Example: Device(config-ecfm-srv)# exit	Returns to Ethernet connectivity fault management configuration mode.
Step 8	mep archive-hold-time <i>minutes</i> Example: Device(config-ecfm)# mep archive-hold-time 65	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	exit Example: Device(config-ecfm)# exit	Returns the CLI to global configuration mode.
Step 10	ethernet cfm global Example: Device(config)# ethernet cfm global	Enables CFM processing globally on the device.
Step 11	interface <i>type number</i> Example: Device(config)# interface gigabitethernet3/1	Specifies an interface and enters interface configuration mode.

	Command or Action	Purpose
Step 12	service instance <i>id</i> ethernet [<i>evc-name</i>] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 13	encapsulation <i>encapsulation-type</i> Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 14	bridge-domain <i>bridge-id</i> Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 15	cfm mip level <i>level</i> Example: Device(config-if-srv)# cfm mip level 4	Creates a MIP and sets the maintenance level number.
Step 16	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Provisioning the Network on the N-PE A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **service** {*short-ma-name* | **number** *MA-number* | **vlan-id** *primary-vlan-id* | **vpn-id** *vpn-id*} {**vlan** *vlan-id* | **port** | **evc** *evc-name*} **direction** {**up** | **down**}
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **ethernet cfm global**
9. **ethernet cfm traceroute cache**
10. **ethernet cfm traceroute cache** **size** *entries*
11. **ethernet cfm traceroute cache** **hold-time** *minutes*
12. **interface** *type number*
13. **service instance** *id* **ethernet** [*evc-name*]
14. **encapsulation** *encapsulation-type*
15. **bridge-domain** *bridge-id*
16. **cfm mip level** *level*
17. **exit**
18. **exit**
19. **snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**]
20. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown** | **mep-missing** | **service-up**]
21. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: Device(config)# ethernet cfm domain Customer level 7	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	service { <i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i> } { vlan <i>vlan-id</i> port evc <i>evc-name</i> } direction { up down } Example: Device(config-ecfm)# service s41 evc 41 vlan 41	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	continuity-check Example: Device(config-ecfm-srv)# continuity-check	Configures the transmission of continuity check messages (CCMs).
Step 6	continuity-check [interval <i>cc-interval</i>] Example: Device(config-ecfm-srv)# continuity-check interval 10s	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	exit Example: Device(config-ecfm-srv)# exit	Returns to Ethernet connectivity fault management configuration mode.
Step 8	ethernet cfm global Example: Device(config)# ethernet cfm global	Enables CFM processing globally on the device.
Step 9	ethernet cfm traceroute cache Example: Device(config)# ethernet cfm traceroute cache	Enables caching of CFM data learned through traceroute messages.
Step 10	ethernet cfm traceroute cache size <i>entries</i> Example: Device(config)# ethernet cfm traceroute cache size 200	Sets the maximum size for the CFM traceroute cache table.

	Command or Action	Purpose
Step 11	ethernet cfm traceroute cache hold-time <i>minutes</i> Example: Device(config)# ethernet cfm traceroute cache hold-time 60	Sets the amount of time that CFM traceroute cache entries are retained.
Step 12	interface <i>type number</i> Example: Device(config)# interface gigabitethernet3/0	Specifies an interface and enters interface configuration mode.
Step 13	service instance <i>id</i> ethernet [<i>evc-name</i>] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 14	encapsulation <i>encapsulation-type</i> Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 15	bridge-domain <i>bridge-id</i> Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 16	cfm mip level <i>level</i> Example: Device(config-if-srv)# cfm mip level 4	Creates a MIP and sets the maintenance level number.
Step 17	exit Example: Device(config-if-srv)# exit	Returns to interface configuration mode.
Step 18	exit Example: Device(config-if)# exit	Returns to global configuration mode.

	Command or Action	Purpose
Step 19	<p>snmp-server enable traps ethernet cfm cc [mep-up] [mep-down] [config] [loop] [cross-connect]</p> <p>Example:</p> <pre>Device(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect</pre>	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.
Step 20	<p>snmp-server enable traps ethernet cfm crosscheck [mep-unknown mep-missing service-up]</p> <p>Example:</p> <pre>Device(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up</pre>	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.
Step 21	<p>end</p> <p>Example:</p> <pre>Device(config)# end</pre>	Returns to privileged EXEC mode.

Provisioning the Network on the CE-B

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
Step 4	<p>service {<i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i>} {vlan <i>vlan-id</i> port evc <i>evc-name</i>} direction {up down}</p> <p>Example: Device(config-ecfm)# service s41 evc 41 vlan 41</p>	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	<p>continuity-check</p> <p>Example: Device(config-ecfm-srv)# continuity-check</p>	Configures the transmission of continuity check messages (CCMs).
Step 6	<p>continuity-check [interval <i>cc-interval</i>]</p> <p>Example: Device(config-ecfm-srv)# continuity-check interval 10s</p>	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	<p>exit</p> <p>Example: Device(config-ecfm-srv)# exit</p>	Returns to Ethernet connectivity fault management configuration mode.
Step 8	<p>mep archive-hold-time <i>minutes</i></p> <p>Example: Device(config-ecfm)# mep archive-hold-time 60</p>	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	<p>exit</p> <p>Example: Device(config-ecfm)# exit</p>	Returns to global configuration mode.
Step 10	<p>ethernet cfm global</p> <p>Example: Device(config)# ethernet cfm global</p>	Enables CFM processing globally on the device.
Step 11	<p>ethernet cfm traceroute cache</p> <p>Example: Device(config)# ethernet cfm traceroute cache</p>	Enables caching of CFM data learned through traceroute messages.
Step 12	<p>ethernet cfm traceroute cache size <i>entries</i></p> <p>Example: Device(config)# ethernet cfm traceroute cache size 200</p>	Sets the maximum size for the CFM traceroute cache table.

	Command or Action	Purpose
Step 13	ethernet cfm traceroute cache hold-time <i>minutes</i> Example: Device(config)# ethernet cfm traceroute cache hold-time 60	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	snmp-server enable traps ethernet cfm cc [mep-up] [mep-down] [config] [loop] [cross-connect] Example: Device(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.
Step 15	snmp-server enable traps ethernet cfm crosscheck [mep-unknown mep-missing service-up] Example: Device(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.
Step 16	end Example: Device(config)# end#	Returns to privileged EXEC mode.

Provisioning the Network on the U-PE B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name level level-id*
4. **service** {*short-ma-name* | **number** *MA-number* | **vlan-id** *primary-vlan-id* | **vpn-id** *vpn-id*} {**vlan** *vlan-id* | **port** | **evc** *evc-name*} **direction** {**up** | **down**}
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
11. **ethernet cfm traceroute cache**
12. **ethernet cfm traceroute cache size** *entries*
13. **ethernet cfm traceroute cache hold-time** *minutes*
14. **interface** *type number*
15. **service instance** *id* **ethernet** [*evc-name*]
16. **encapsulation** *encapsulation-type*
17. **bridge-domain** *bridge-id*
18. **cfm mip level** *level*
19. **exit**
20. **exit**
21. **snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**]
22. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown** | **mep-missing** | **service-up**]
23. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: Device(config)# ethernet cfm domain Customer level 7	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	service { <i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i> } { vlan <i>vlan-id</i> port evc <i>evc-name</i> } direction { up down } Example: Device(config-ecfm)# service s41 evc 41 vlan 41	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	continuity-check Example: Device(config-ecfm-srv)# continuity-check	Configures the transmission of continuity check messages (CCMs).
Step 6	continuity-check [interval <i>cc-interval</i>] Example: Device(config-ecfm-srv)# continuity-check interval 10s	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	exit Example: Device(config-ecfm-srv)# exit	Returns to Ethernet connectivity fault management configuration mode.
Step 8	mep archive-hold-time <i>minutes</i> Example: Device(config-ecfm)# mep archive-hold-time 60	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	exit Example: Device(config-ecfm)# exit	Returns to global configuration mode.
Step 10	ethernet cfm global Example: Device(config)# ethernet cfm global	Enables CFM processing globally on the device.
Step 11	ethernet cfm traceroute cache Example: Device(config)# ethernet cfm traceroute cache	Enables caching of CFM data learned through traceroute messages.

	Command or Action	Purpose
Step 12	ethernet cfm traceroute cache size <i>entries</i> Example: Device(config)# ethernet cfm traceroute cache size 200	Sets the maximum size for the CFM traceroute cache table.
Step 13	ethernet cfm traceroute cache hold-time <i>minutes</i> Example: Device(config)# ethernet cfm traceroute cache hold-time 60	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	interface <i>type number</i> Example: Device(config)# interface gigabitethernet2/0	Specifies an interface and enters interface configuration mode.
Step 15	service instance <i>id</i> ethernet [<i>evc-name</i>] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	encapsulation <i>encapsulation-type</i> Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 17	bridge-domain <i>bridge-id</i> Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 18	cfm mip level <i>level</i> Example: Device(config-if-srv)# cfm mip level 4	Creates a MIP and sets the maintenance level number.
Step 19	exit Example: Device(config-if-srv)# exit	Returns to interface configuration mode.
Step 20	exit Example: Device(config-if)# exit	Returns to global configuration mode.

	Command or Action	Purpose
Step 21	snmp-server enable traps ethernet cfm cc [mep-up] [mep-down] [config] [loop] [cross-connect] Example: Device(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.
Step 22	snmp-server enable traps ethernet cfm crosscheck [mep-unknown mep-missing service-up] Example: Device(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.
Step 23	end Example: Device(config)# end	Returns to privileged EXEC mode.

Provisioning the Network on the PE-AGG B

SUMMARY STEPS

1. enable
2. configure terminal
3. ethernet cfm domain *domain-name* level *level-id*
4. service {*short-ma-name* | number *MA-number* | **vlan-id** *primary-vlan-id* | **vpn-id** *vpn-id*} {**vlan** *vlan-id* | **port** | **evc** *evc-name*} **direction** {**up** | **down**}
5. continuity-check
6. continuity-check [**interval** *cc-interval*]
7. exit
8. mep archive-hold-time *minutes*
9. exit
10. ethernet cfm global
11. interface *type number*
12. service instance *id* ethernet [*evc-name*]
13. encapsulation *encapsulation-type*
14. bridge-domain *bridge-id*
15. cfm mip level *level*
16. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: Device(config)# ethernet cfm domain Customer level 7	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	service { <i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i> } { vlan <i>vlan-id</i> port evc <i>evc-name</i> } direction { up down } Example: Device(config-ecfm)# service s41 evc 41 vlan 41	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	continuity-check Example: Device(config-ecfm-srv)# continuity-check	Configures the transmission of continuity check messages (CCMs).
Step 6	continuity-check [interval <i>cc-interval</i>] Example: Device(config-ecfm-srv)# continuity-check interval 10s	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	exit Example: Device(config-ecfm-srv)# exit	Returns to Ethernet connectivity fault management configuration mode.
Step 8	mep archive-hold-time <i>minutes</i> Example: Device(config-ecfm)# mep archive-hold-time 65	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	exit Example: Device(config-ecfm)# exit	Returns to global configuration mode.

	Command or Action	Purpose
Step 10	ethernet cfm global Example: Device(config)# ethernet cfm global	Enables CFM processing globally on the device.
Step 11	interface <i>type number</i> Example: Device(config)# interface gigabitethernet1/1	Specifies an interface and enters interface configuration mode.
Step 12	service instance <i>id</i> ethernet [<i>evc-name</i>] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 13	encapsulation <i>encapsulation-type</i> Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 14	bridge-domain <i>bridge-id</i> Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 15	cfm mip level <i>level</i> Example: Device(config-if-srv)# cfm mip level 4	Creates a MIP and sets the maintenance level number.
Step 16	end Example: Device(config-if-srv)# end	Returns to privileged EXEC mode.

Provisioning the Network on the N-PE B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **service** {*short-ma-name* | **number** *MA-number* | **vlan-id** *primary-vlan-id* | **vpn-id** *vpn-id*} {**vlan** *vlan-id* | **port** | **evc** *evc-name*} **direction** {**up** | **down**}
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
11. **ethernet cfm traceroute cache**
12. **ethernet cfm traceroute cache** **size** *entries*
13. **ethernet cfm traceroute cache** **hold-time** *minutes*
14. **interface** *type number*
15. **service instance** *id* **ethernet** [*evc-name*]
16. **encapsulation** *encapsulation-type*
17. **bridge-domain** *bridge-id*
18. **cfm mip level** *level*
19. **exit**
20. **exit**
21. **snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**]
22. **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown** | **mep-missing** | **service-up**]
23. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: Device(config)# ethernet cfm domain Customer level 7	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	service { <i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i> } { vlan <i>vlan-id</i> port evc <i>evc-name</i> } direction { up down } Example: Device(config-ecfm)# service s41 evc 41 vlan 41	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	continuity-check Example: Device(config-ecfm-srv)# continuity-check	Configures the transmission of continuity check messages (CCMs).
Step 6	continuity-check [interval <i>cc-interval</i>] Example: Device(config-ecfm-srv)# continuity-check interval 10s	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	exit Example: Device(config-ecfm-srv)# exit	Returns to Ethernet connectivity fault management configuration mode.
Step 8	mep archive-hold-time <i>minutes</i> Example: Device(config-ecfm)# mep archive-hold-time 60	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	exit Example: Device(config-ecfm)# exit	Returns to global configuration mode.
Step 10	ethernet cfm global Example: Device(config)# ethernet cfm global	Enables CFM processing globally on the device.
Step 11	ethernet cfm traceroute cache Example: Device(config)# ethernet cfm traceroute cache	Enables caching of CFM data learned through traceroute messages.

	Command or Action	Purpose
Step 12	ethernet cfm traceroute cache size <i>entries</i> Example: Device(config)# ethernet cfm traceroute cache size 200	Sets the maximum size for the CFM traceroute cache table.
Step 13	ethernet cfm traceroute cache hold-time <i>minutes</i> Example: Device(config)# ethernet cfm traceroute cache hold-time 60	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	interface <i>type number</i> Example: Device(config)# interface gigabitethernet1/2	Specifies an interface and enters interface configuration mode.
Step 15	service instance <i>id</i> ethernet [<i>evc-name</i>] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	encapsulation <i>encapsulation-type</i> Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 17	bridge-domain <i>bridge-id</i> Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 18	cfm mip level <i>level</i> Example: Device(config-if-srv)# cfm mip level 4	Creates a MIP and sets the maintenance level number.
Step 19	exit Example: Device(config-if-srv)# exit	Returns to interface configuration mode.
Step 20	exit Example: Device(config-if)# exit	Returns to global configuration mode.

	Command or Action	Purpose
Step 21	snmp-server enable traps ethernet cfm cc [mep-up] [mep-down] [config] [loop] [cross-connect] Example: Device(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect	Enables SNMP trap generation for Ethernet CFM mep-up, mep-down, config, loop, and cross-connect events.
Step 22	snmp-server enable traps ethernet cfm crosscheck [mep-unknown mep-missing service-up] Example: Device(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up	Enables SNMP trap generation for Ethernet CFM mep-unknown, mep-missing, and service-up continuity check events in relation to the cross-check operation between statically configured MEPs and those learned via CCMs.
Step 23	end Example: Device(config)# end	Returns to privileged EXEC mode.

Provisioning Service

Provisioning Service on the CE-A

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **service** {*short-ma-name* | **number** *MA-number* | **vlan-id** *primary-vlan-id* | **vpn-id** *vpn-id*} {**vlan** *vlan-id* | **port** | **evc** *evc-name*} **direction** {**up** | **down**}
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
11. **ethernet cfm traceroute cache**
12. **ethernet cfm traceroute cache** **size** *entries*
13. **ethernet cfm traceroute cache** **hold-time** *minutes*
14. **interface** *type number*
15. **service instance** *id* **ethernet** [*evc-name*]
16. **encapsulation** *encapsulation-type*
17. **bridge-domain** *bridge-id*
18. **cfm mep domain** *domain-name* **mpid** *id*
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: Device(config)# ethernet cfm domain Customer level 7	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 4	service { <i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i> } { vlan <i>vlan-id</i> port evc <i>evc-name</i> } direction { up down } Example: Device(config-ecfm)# service s41 evc 41 vlan 41	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	continuity-check Example: Device(config-ecfm-srv)# continuity-check	Configures the transmission of continuity check messages (CCMs).
Step 6	continuity-check [interval <i>cc-interval</i>] Example: Device(config-ecfm-srv)# continuity-check interval 10s	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	exit Example: Device(config-ecfm-srv)# exit	Returns to Ethernet connectivity fault management configuration mode.
Step 8	mep archive-hold-time <i>minutes</i> Example: Device(config-ecfm)# mep archive-hold-time 60	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	exit Example: Device(config-ecfm)# exit	Returns to global configuration mode.
Step 10	ethernet cfm global Example: Device(config)# ethernet cfm global	Enables CFM processing globally on the device.
Step 11	ethernet cfm traceroute cache Example: Device(config)# ethernet cfm traceroute cache	Enables caching of CFM data learned through traceroute messages.
Step 12	ethernet cfm traceroute cache size <i>entries</i> Example: Device(config)# ethernet cfm traceroute cache size 200	Sets the maximum size for the CFM traceroute cache table.

	Command or Action	Purpose
Step 13	ethernet cfm traceroute cache hold-time minutes Example: Device(config)# ethernet cfm traceroute cache hold-time 60	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	interface type number Example: Device(config)# interface ethernet 0/3	Specifies an interface and enters interface configuration mode.
Step 15	service instance id ethernet [evc-name] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	encapsulation encapsulation-type Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 17	bridge-domain bridge-id Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 18	cfm mep domain domain-name mpid id Example: Device(config-if-srv)# cfm mep domain L4 mpid 4001	Configures the MEP domain and the ID.
Step 19	end Example: Device(config-if-srv)# end	Returns to privileged EXEC mode.

Provisioning Service on the U-PE A

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: Device(config)# ethernet cfm domain Customer level 7	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	service { <i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i> } { vlan <i>vlan-id</i> port evc <i>evc-name</i> } direction { up down } Example: Device(config-ecfm)# service s41 evc 41 vlan 41	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	continuity-check Example: Device(config-ecfm-srv)# continuity-check	Configures the transmission of continuity check messages (CCMs).
Step 6	continuity-check [interval <i>cc-interval</i>] Example: Device(config-ecfm-srv)# continuity-check interval 10s	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	exit Example: Device(config-ecfm-srv)# exit	Returns to Ethernet connectivity fault management configuration mode.
Step 8	mep archive-hold-time <i>minutes</i> Example: Device(config-ecfm)# mep archive-hold-time 60	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	exit Example: Device(config-ecfm)# exit	Returns to global configuration mode.
Step 10	ethernet cfm global Example: Device(config)# ethernet cfm global	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 11	ethernet cfm traceroute cache Example: Device(config)# ethernet cfm traceroute cache	Enables caching of CFM data learned through traceroute messages.
Step 12	ethernet cfm traceroute cache size entries Example: Device(config)# ethernet cfm traceroute cache size 200	Sets the maximum size for the CFM traceroute cache table.
Step 13	ethernet cfm traceroute cache hold-time minutes Example: Device(config)# ethernet cfm traceroute cache hold-time 60	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	interface type number Example: Device(config)# interface gigabitethernet3/2	Specifies an interface and enters interface configuration mode.
Step 15	service instance id ethernet [evc-name] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	encapsulation encapsulation-type Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 17	bridge-domain bridge-id Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 18	cfm mep domain domain-name mpid id Example: Device(config-if-srv)# cfm mep domain L4 mpid 4001	Configures the MEP domain and the ID.
Step 19	exit Example: Device(config-if-srv)# exit	Returns to interface configuration mode.

	Command or Action	Purpose
Step 20	exit Example: Device(config-if)# exit	Returns to global configuration mode.
Step 21	interface <i>type number</i> Example: Device(config)# interface gigabitethernet3/2	Specifies an interface and enters interface configuration mode.
Step 22	service instance <i>id ethernet</i> [<i>evc-name</i>] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 23	encapsulation <i>encapsulation-type</i> Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 24	bridge-domain <i>bridge-id</i> Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 25	cfm mip level <i>level</i> Example: Device(config-if-srv)#cfm mip level 4	Creates a MIP and sets the maintenance level number.
Step 26	end Example: Device(config-if-srv)# end	Returns to privileged EXEC mode.

Provisioning Service on the PE-AGG A

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **service** {*short-ma-name* | **number** *MA-number* | **vlan-id** *primary-vlan-id* | **vpn-id** *vpn-id*} {**vlan** *vlan-id* | **port** | **evc** *evc-name*} **direction** {**up** | **down**}
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
11. **interface** *type number*
12. **service instance** *id* **ethernet** [*evc-name*]
13. **encapsulation** *encapsulation-type*
14. **bridge-domain** *bridge-id*
15. **cfm mip level** *level*
16. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 4	<p>service {<i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i>} {vlan <i>vlan-id</i> port evc <i>evc-name</i>} direction {up down}</p> <p>Example: Device(config-ecfm)# service s41 evc 41 vlan 41</p>	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	<p>continuity-check</p> <p>Example: Device(config-ecfm-srv)# continuity-check</p>	Configures the transmission of continuity check messages (CCMs).
Step 6	<p>continuity-check [interval <i>cc-interval</i>]</p> <p>Example: Device(config-ecfm-srv)# continuity-check interval 10s</p>	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	<p>exit</p> <p>Example: Device(config-ecfm-srv)# exit</p>	Returns to Ethernet connectivity fault management configuration mode.
Step 8	<p>mep archive-hold-time <i>minutes</i></p> <p>Example: Device(config-ecfm)# mep archive-hold-time 65</p>	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	<p>exit</p> <p>Example: Device(config-ecfm)# exit</p>	Returns to global configuration mode.
Step 10	<p>ethernet cfm global</p> <p>Example: Device(config)# ethernet cfm global</p>	Enables CFM processing globally on the device.
Step 11	<p>interface <i>type number</i></p> <p>Example: Device(config)# interface gigabitethernet3/1</p>	Specifies an interface and enters interface configuration mode.
Step 12	<p>service instance <i>id</i> ethernet [<i>evc-name</i>]</p> <p>Example: Device(config-if)# service instance 333 ethernet evc1</p>	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.

	Command or Action	Purpose
Step 13	encapsulation <i>encapsulation-type</i> Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 14	bridge-domain <i>bridge-id</i> Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 15	cfm mip level <i>level</i> Example: Device(config-if-srv)#cfm mip level 4	Creates a MIP and sets the maintenance level number.
Step 16	end Example: Device(config-if-srv)# end	Returns to privileged EXEC mode.

Provisioning Service on the N-PE A

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: Device(config)# ethernet cfm domain Customer level 7	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	service { <i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i> } { vlan <i>vlan-id</i> port evc <i>evc-name</i> } direction { up down } Example: Device(config-ecfm)# service s41 evc 41 vlan 41	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	continuity-check Example: Device(config-ecfm-srv)# continuity-check	Configures the transmission of continuity check messages (CCMs).
Step 6	continuity-check [interval <i>cc-interval</i>] Example: Device(config-ecfm-srv)# continuity-check interval 10s	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	exit Example: Device(config-ecfm-srv)# exit	Returns to Ethernet connectivity fault management configuration mode.
Step 8	mep archive-hold-time <i>minutes</i> Example: Device(config-ecfm)# mep archive-hold-time 60	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	exit Example: Device(config-ecfm)# exit	Returns to global configuration mode.
Step 10	ethernet cfm global Example: Device(config)# ethernet cfm global	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 11	ethernet cfm traceroute cache Example: Device(config)# ethernet cfm traceroute cache	Enables caching of CFM data learned through traceroute messages.
Step 12	ethernet cfm traceroute cache size entries Example: Device(config)# ethernet cfm traceroute cache size 200	Sets the maximum size for the CFM traceroute cache table.
Step 13	ethernet cfm traceroute cache hold-time minutes Example: Device(config)# ethernet cfm traceroute cache hold-time 60	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	interface type number Example: Device(config)# interface gigabitethernet3/0	Specifies an interface and enters interface configuration mode.
Step 15	service instance id ethernet [evc-name] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	encapsulation encapsulation-type Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 17	bridge-domain bridge-id Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 18	cfm mip level level Example: Device(config-if-srv)# cfm mip level 4	Creates a MIP and sets the maintenance level number.
Step 19	exit Example: Device(config-if-srv)# exit	Returns to interface configuration mode.

	Command or Action	Purpose
Step 20	exit Example: Device(config-if)# exit	Returns to global configuration mode.
Step 21	interface <i>type number</i> Example: Device(config-if)# interface gigabitethernet4/0	Specifies an interface.
Step 22	service instance <i>id ethernet</i> [<i>evc-name</i>] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 23	encapsulation <i>encapsulation-type</i> Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 24	bridge-domain <i>bridge-id</i> Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 25	cfm mep domain <i>domain-name mpid id</i> Example: Device(config-if-srv)# cfm mep domain L4 mpid 4001	Configures the MEP domain and the ID.
Step 26	end Example: Device(config-if-srv)# end	Returns to privileged EXEC mode.

Provisioning Service on the CE-B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name* **level** *level-id*
4. **service** {*short-ma-name* | **number** *MA-number* | **vlan-id** *primary-vlan-id* | **vpn-id** *vpn-id*} {**vlan** *vlan-id* | **port** | **evc** *evc-name*} **direction** {**up** | **down**}
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
11. **ethernet cfm traceroute cache**
12. **ethernet cfm traceroute cache** **size** *entries*
13. **ethernet cfm traceroute cache** **hold-time** *minutes*
14. **interface** *type number*
15. **service instance** *id* **ethernet** [*evc-name*]
16. **encapsulation** *encapsulation-type*
17. **bridge-domain** *bridge-id*
18. **cfm mep domain** *domain-name* **mpid** *id*
19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: Device(config)# ethernet cfm domain Customer level 7	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 4	service { <i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i> } { vlan <i>vlan-id</i> port evc <i>evc-name</i> } direction { up down } Example: Device(config-ecfm)# service s41 evc 41 vlan 41	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	continuity-check Example: Device(config-ecfm-srv)# continuity-check	Configures the transmission of continuity check messages (CCMs).
Step 6	continuity-check [interval <i>cc-interval</i>] Example: Device(config-ecfm-srv)# continuity-check interval 10s	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	exit Example: Device(config-ecfm-srv)# exit	Returns to Ethernet connectivity fault management configuration mode.
Step 8	mep archive-hold-time <i>minutes</i> Example: Device(config-ecfm)# mep archive-hold-time 60	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	exit Example: Device(config-ecfm)# exit	Returns to global configuration mode.
Step 10	ethernet cfm global Example: Device(config)# ethernet cfm global	Enables CFM processing globally on the device.
Step 11	ethernet cfm traceroute cache Example: Device(config)# ethernet cfm traceroute cache	Enables caching of CFM data learned through traceroute messages.
Step 12	ethernet cfm traceroute cache size <i>entries</i> Example: Device(config)# ethernet cfm traceroute cache size 200	Sets the maximum size for the CFM traceroute cache table.

	Command or Action	Purpose
Step 13	ethernet cfm traceroute cache hold-time minutes Example: Device(config)# ethernet cfm traceroute cache hold-time 60	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	interface type number Example: Device(config)# interface ethernet 0/1	Specifies an interface and enters interface configuration mode.
Step 15	service instance id ethernet [evc-name] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	encapsulation encapsulation-type Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 17	bridge-domain bridge-id Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 18	cfm mep domain domain-name mpid id Example: Device(config-if-srv)# cfm mep domain L4 mpid 4001	Configures the MEP domain and the ID.
Step 19	end Example: Device(config-if-srv)# end	Returns to privileged EXEC mode.

Provisioning Service on the U-PE B

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ethernet cfm domain domain-name level level-id Example: Device(config)# ethernet cfm domain Customer level 7	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	service {short-ma-name number MA-number vlan-id primary-vlan-id vpn-id vpn-id} {vlan vlan-id port evc evc-name} direction {up down} Example: Device(config-ecfm)# service s41 evc 41 vlan 41	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	continuity-check Example: Device(config-ecfm-srv)# continuity-check	Configures the transmission of continuity check messages (CCMs).
Step 6	continuity-check [interval cc-interval] Example: Device(config-ecfm-srv)# continuity-check interval 10s	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	exit Example: Device(config-ecfm-srv)# exit	Returns to Ethernet connectivity fault management configuration mode.
Step 8	mep archive-hold-time minutes Example: Device(config-ecfm)# mep archive-hold-time 60	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	exit Example: Device(config-ecfm)# exit	Returns to global configuration mode.
Step 10	ethernet cfm global Example: Device(config)# ethernet cfm global	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 11	ethernet cfm traceroute cache Example: Device(config)# ethernet cfm traceroute cache	Enables caching of CFM data learned through traceroute messages.
Step 12	ethernet cfm traceroute cache size entries Example: Device(config)# ethernet cfm traceroute cache size 200	Sets the maximum size for the CFM traceroute cache table.
Step 13	ethernet cfm traceroute cache hold-time minutes Example: Device(config)# ethernet cfm traceroute cache hold-time 60	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	interface type number Example: Device(config)# interface gigabitethernet1/0	Specifies an interface and enters interface configuration mode.
Step 15	service instance id ethernet [evc-name] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	encapsulation encapsulation-type Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 17	bridge-domain bridge-id Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 18	cfm mip level level Example: Device(config-if-srv)# cfm mip level 4	Creates a MIP and sets the maintenance level number.
Step 19	exit Example: Device(config-if-srv)# exit	Returns to interface configuration mode.

	Command or Action	Purpose
Step 20	exit Example: Device(config-if)# exit	Returns to global configuration mode.
Step 21	interface <i>type number</i> Example: Device(config)# interface gigabitethernet1/0	Specifies an interface and enters interface configuration mode.
Step 22	service instance <i>id ethernet</i> [<i>evc-name</i>] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 23	encapsulation <i>encapsulation-type</i> Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 24	bridge-domain <i>bridge-id</i> Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 25	cfm mep domain <i>domain-name mpid id</i> Example: Device(config-if-srv)# cfm mep domain L4 mpid 4001	Configures the MEP domain and the ID.
Step 26	end Example: Device(config-if-srv)# end	Returns to privileged EXEC mode.

Provisioning Service on the PE-AGG B

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ethernet cfm domain** *domain-name level level-id*
4. **service** {*short-ma-name* | **number** *MA-number* | **vlan-id** *primary-vlan-id* | **vpn-id** *vpn-id*} {**vlan** *vlan-id* | **port** | **evc** *evc-name*} **direction** {**up** | **down**}
5. **continuity-check**
6. **continuity-check** [**interval** *cc-interval*]
7. **exit**
8. **mep archive-hold-time** *minutes*
9. **exit**
10. **ethernet cfm global**
11. **interface** *type number*
12. **service instance** *id* **ethernet** [*evc-name*]
13. **encapsulation** *encapsulation-type*
14. **bridge-domain** *bridge-id*
15. **cfm mip level** *level*
16. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 4	<p>service {<i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i>} {vlan <i>vlan-id</i> port evc <i>evc-name</i>} direction {up down}</p> <p>Example: Device(config-ecfm)# service s41 evc 41 vlan 41</p>	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	<p>continuity-check</p> <p>Example: Device(config-ecfm-srv)# continuity-check</p>	Configures the transmission of continuity check messages (CCMs).
Step 6	<p>continuity-check [interval <i>cc-interval</i>]</p> <p>Example: Device(config-ecfm-srv)# continuity-check interval 10s</p>	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	<p>exit</p> <p>Example: Device(config-ecfm-srv)# exit</p>	Returns to Ethernet connectivity fault management configuration mode.
Step 8	<p>mep archive-hold-time <i>minutes</i></p> <p>Example: Device(config-ecfm)# mep archive-hold-time 65</p>	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	<p>exit</p> <p>Example: Device(config-ecfm)# exit</p>	Returns to global configuration mode.
Step 10	<p>ethernet cfm global</p> <p>Example: Device(config)# ethernet cfm global</p>	Enables CFM processing globally on the device.
Step 11	<p>interface <i>type number</i></p> <p>Example: Device(config)# interface gigabitethernet3/1</p>	Specifies an interface and enters interface configuration mode.
Step 12	<p>service instance <i>id</i> ethernet [<i>evc-name</i>]</p> <p>Example: Device(config-if)# service instance 333 ethernet evc1</p>	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.

	Command or Action	Purpose
Step 13	encapsulation <i>encapsulation-type</i> Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 14	bridge-domain <i>bridge-id</i> Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 15	cfm mip level <i>level</i> Example: Device(config-if-srv)#cfm mip level 4	Creates a MIP and sets the maintenance level number.
Step 16	end Example: Device(config-if-srv)# end	Returns to privileged EXEC mode.

Provisioning Service on the N-PE B

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: Device(config)# ethernet cfm domain Customer level 7	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	service { <i>short-ma-name</i> number <i>MA-number</i> vlan-id <i>primary-vlan-id</i> vpn-id <i>vpn-id</i> } { vlan <i>vlan-id</i> port evc <i>evc-name</i> } direction { up down } Example: Device(config-ecfm)# service s41 evc 41 vlan 41	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.
Step 5	continuity-check Example: Device(config-ecfm-srv)# continuity-check	Configures the transmission of continuity check messages (CCMs).
Step 6	continuity-check [interval <i>cc-interval</i>] Example: Device(config-ecfm-srv)# continuity-check interval 10s	Configures the per-service parameters and sets the interval at which CCMs are transmitted.
Step 7	exit Example: Device(config-ecfm-srv)# exit	Returns to Ethernet connectivity fault management configuration mode.
Step 8	mep archive-hold-time <i>minutes</i> Example: Device(config-ecfm)# mep archive-hold-time 60	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	exit Example: Device(config-ecfm)# exit	Returns to global configuration mode.
Step 10	ethernet cfm global Example: Device(config)# ethernet cfm global	Enables CFM processing globally on the device.

	Command or Action	Purpose
Step 11	ethernet cfm traceroute cache Example: Device(config)# ethernet cfm traceroute cache	Enables caching of CFM data learned through traceroute messages.
Step 12	ethernet cfm traceroute cache size entries Example: Device(config)# ethernet cfm traceroute cache size 200	Sets the maximum size for the CFM traceroute cache table.
Step 13	ethernet cfm traceroute cache hold-time minutes Example: Device(config)# ethernet cfm traceroute cache hold-time 60	Sets the amount of time that CFM traceroute cache entries are retained.
Step 14	interface type number Example: Device(config)# interface gigabitethernet3/0	Specifies an interface and enters interface configuration mode.
Step 15	service instance id ethernet [evc-name] Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 16	encapsulation encapsulation-type Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 17	bridge-domain bridge-id Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 18	cfm mip level level Example: Device(config-if-srv)# cfm mip level 4	Creates a MIP and sets the maintenance level number.
Step 19	exit Example: Device(config-if-srv)# exit	Returns to interface configuration mode.

	Command or Action	Purpose
Step 20	exit Example: Device(config-if)# exit	Returns to global configuration mode.
Step 21	interface <i>type number</i> Example: Device(config-if)# interface gigabitethernet4/0	Specifies an interface.
Step 22	service instance <i>id ethernet [evc-name]</i> Example: Device(config-if)# service instance 333 ethernet evc1	Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.
Step 23	encapsulation <i>encapsulation-type</i> Example: Device(config-if-srv)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 24	bridge-domain <i>bridge-id</i> Example: Device(config-if-srv)# bridge-domain 100	Binds a service instance to a bridge domain instance.
Step 25	cfm mep domain <i>domain-name mpid id</i> Example: Device(config-if-srv)# cfm mep domain L4 mpid 4001	Configures the MEP domain and the ID.
Step 26	end Example: Device(config-if-srv)# end	Returns to privileged EXEC mode.

Configuring and Enabling the Cross-Check Function

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: Device(config)# ethernet cfm domain ServiceProvider level 4	Defines a CFM domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	mep crosscheck mpid <i>id</i> vlan <i>vlan-id</i> [mac <i>mac-address</i>] Example: Device(config-ether-cfm)# mep crosscheck mpid 402 vlan 100	Statically defines a remote MEP on a specified VLAN within the domain.
Step 5	exit Example: Device(config-ether-cfm)# exit#	Returns to global configuration mode.
Step 6	ethernet cfm mep crosscheck start-delay <i>delay</i> Example: Device(config)# ethernet cfm mep crosscheck start-delay 60	Configures the maximum amount of time that the device waits for remote MEPs to come up before the cross-check operation is started

	Command or Action	Purpose
Step 7	exit Example: Device(config)# exit	Returns to privileged EXEC mode.
Step 8	ethernet cfm mep crosscheck {enable disable} level {level-id level-id-level-id [,level-id-level-id]} vlan {vlan-id any vlan-id-vlan-id [,vlan-id-vlan-id]} Example: Device# ethernet cfm mep crosscheck enable level 4 vlan 100	Enables cross-checking between remote MEPs in the domain and MEPs learned through CCMs.

Example

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

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

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

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

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: Device(config)# ethernet cfm domain ServiceProvider level 4	Defines a CFM domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	mep crosscheck mpid <i>id</i> vlan <i>vlan-id</i> [mac <i>mac-address</i>] Example: Device(config-ether-cfm)# mep crosscheck mpid 401 vlan 100	Statically defines a remote MEP on a specified VLAN within the domain.
Step 5	exit Example: Device(config-ether-cfm)# exit	Returns to global configuration mode.
Step 6	ethernet cfm mep crosscheck start-delay <i>delay</i> Example: Device(config)# ethernet cfm mep crosscheck start-delay 60	Configures the maximum amount of time that the device waits for remote MEPs to come up before the cross-check operation is started.
Step 7	exit Example: Device(config)# exit	Returns to privileged EXEC mode.
Step 8	ethernet cfm mep crosscheck { enable disable } level { <i>level-id</i> <i>level-id-level-id</i> [, <i>level-id-level-id</i>]} vlan { <i>vlan-id</i> any <i>vlan-id-vlan-id</i> [, <i>vlan-id-vlan-id</i>]} Example: Device# ethernet cfm mep crosscheck enable level 4 vlan 100	Enables cross-checking between MEPs.

Example

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

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

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

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

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

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

DETAILED STEPS

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

	Command or Action	Purpose
Step 4	mep crosscheck mpid <i>id</i> vlan <i>vlan-id</i> [mac <i>mac-address</i>] Example: Device(config-ether-cfm)# mep crosscheck mpid 702 vlan 100	Statically defines a remote MEP with a specified ID, VLAN, and domain.
Step 5	exit Example: Device(config-ether-cfm)# exit	Returns to global configuration mode.
Step 6	ethernet cfm mep crosscheck start-delay <i>delay</i> Example: Device(config)# ethernet cfm mep crosscheck start-delay 60	Configures the maximum amount of time that the device waits for remote MEPs to come up before the cross-check operation is started.
Step 7	exit Example: Device(config)# exit	Returns to privileged EXEC mode.
Step 8	ethernet cfm mep crosscheck { enable disable } level { <i>level-id</i> <i>level-id-level-id</i> [, <i>level-id-level-id</i>]} vlan { <i>vlan-id</i> any <i>vlan-id-vlan-id</i> [, <i>vlan-id-vlan-id</i>]} Example: Device# ethernet cfm mep crosscheck enable level 7 vlan 100	Enables cross-checking between MEPs.

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> [direction outward] Example: Device(config)# ethernet cfm domain Customer level 7 direction outward	Defines an outward CFM domain at a specified level and enters Ethernet CFM configuration mode.
Step 4	mep crosscheck mpid <i>id</i> vlan <i>vlan-id</i> [mac <i>mac-address</i>] Example: Device(config-ether-cfm)# mep crosscheck mpid 401 vlan 100	Statically defines a remote MEP on a VLAN within a specified domain.
Step 5	exit Example: Device(config-ether-cfm)# exit	Returns to global configuration mode.
Step 6	ethernet cfm mep crosscheck start-delay <i>delay</i> Example: Device(config)# ethernet cfm mep crosscheck start-delay 60	Configures the maximum amount of time that the device waits for remote MEPs to come up before the cross-check operation is started.
Step 7	exit Example: Device(config)# exit	Returns to privileged EXEC mode.
Step 8	ethernet cfm mep crosscheck { enable disable } level { <i>level-id</i> <i>level-id-level-id</i> [, <i>level-id-level-id</i>]} vlan { <i>vlan-id</i> any <i>vlan-id-vlan-id</i> [, <i>vlan-id-vlan-id</i>]} Example: Device# ethernet cfm mep crosscheck enable level 7 vlan 100	Enables cross-checking between MEPs.

Configuring CFM over Bridge Domains

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> direction outward Example: Device(config)# ethernet cfm domain CUSTOMER level 7 direction outward	Defines a CFM maintenance domain at a particular level and enters Ethernet CFM configuration mode.
Step 4	service <i>csi-id</i> evc <i>evc-name</i> Example: Device(config-ether-cfm)# service customer_100 evc evc_100	Sets a universally unique ID for a CSI within a maintenance domain.
Step 5	exit Example: Device(config-ether-cfm)# exit	Returns to global configuration mode.
Step 6	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: Device(config)# ethernet cfm domain MIP level 7	Defines a CFM maintenance domain at a particular level and enters Ethernet CFM configuration mode.
Step 7	exit Example: Device(config-ether-cfm)# exit	Returns to global configuration mode.
Step 8	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> Example: Device(config)# ethernet cfm domain PROVIDER level 4	Defines a CFM maintenance domain at a particular level and enters Ethernet CFM configuration mode.

	Command or Action	Purpose
Step 9	<p>service <i>csi-id</i> evc <i>evc-name</i></p> <p>Example:</p> <pre>Device(config-ether-cfm)# service provider_1 evc evc_100</pre>	Sets a universally unique ID for a CSI within a maintenance domain.
Step 10	<p>mep crosscheck mpid <i>id</i> evc <i>evc-name</i> mac <i>mac-address</i></p> <p>Example:</p> <pre>Device(config-ether-cfm)# mep crosscheck mpid 200 evc evc_100 mac 1010.1010.1010</pre>	Statically defines a remote MEP within a maintenance domain.
Step 11	<p>exit</p> <p>Example:</p> <pre>Device(config-ether-cfm)# exit</pre>	Returns to global configuration mode.
Step 12	<p>ethernet evc <i>evc-name</i></p> <p>Example:</p> <pre>Device(config)# ethernet evc evc_100</pre>	Defines an EVC and enters EVC configuration mode.
Step 13	<p>exit</p> <p>Example:</p> <pre>Device(config-evc)# exit</pre>	Returns to global configuration mode.
Step 14	<p>interface <i>type number</i></p> <p>Example:</p> <pre>Device(config)# interface Ethernet 1/0</pre>	Specifies an interface and enters interface configuration mode.
Step 15	<p>no ip address</p> <p>Example:</p> <pre>Device(config-if)# no ip address</pre>	Disables IP processing.
Step 16	<p>service instance <i>id</i> ethernet <i>evc-id</i></p> <p>Example:</p> <pre>Device(config-if)# service instance 100 ethernet evc_100</pre>	Specifies an Ethernet service instance on an interface and enters service instance configuration mode.

	Command or Action	Purpose
Step 17	encapsulation dot1q <i>vlan-id</i> Example: Device(config-if-srv)# encapsulation dot1q 100	Defines the matching criteria to map 802.1Q frames on an ingress interface to the appropriate service instance.
Step 18	bridge-domain <i>bridge-id</i> Example: Device(config-if-srv)# bridge-domain 100	Establishes a bridge domain.
Step 19	cfm mep domain <i>domain-name</i> outward mpid <i>mpid-value</i> Example: Device(config-if-srv)# cfm mep domain CUSTOMER outward mpid 1001	Configures a MEP for a domain.
Step 20	end Example: Device(config-if-srv)# end	Returns to privileged EXEC mode.
Step 21	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 22	interface <i>type name</i> Example: Device(config)# interface Ethernet 1/1	Specifies an interface and enters interface configuration mode.
Step 23	no ip address Example: Device(config-if)# no ip address	Disables IP processing.
Step 24	ethernet cfm mip level <i>level-id</i> Example: Device(config-if)# ethernet cfm mip level 7	Provisions a MIP at a specified maintenance level on an interface.

	Command or Action	Purpose
Step 25	service instance <i>id</i> ethernet <i>evc-id</i> Example: <pre>Device(config-if)# service instance 100 ethernet evc_100</pre>	Configures an Ethernet service instance on an interface and enters service instance configuration mode.
Step 26	encapsulation dot1q <i>vlan-id</i> Example: <pre>Device(config-if-srv)# encapsulation dot1q 100</pre>	Defines the matching criteria to map 802.1Q frames on an ingress interface to the appropriate service instance.
Step 27	bridge-domain <i>bridge-id</i> Example: <pre>Device(config-if-srv)# bridge-domain 100</pre>	Establishes a bridge domain.
Step 28	cfm mep domain <i>domain-name</i> inward mpid <i>mpid-value</i> Example: <pre>Device(config-if-srv)# cfm mep domain PROVIDER inward mpid 201</pre>	Configures a MEP for a domain.
Step 29	end Example: <pre>Device(config-if-srv)# end</pre>	Returns to privileged EXEC mode.
Step 30	configure terminal Example: <pre>Device# configure terminal</pre>	Enters global configuration mode.
Step 31	ethernet cfm cc enable level <i>level-id</i> evc <i>evc-name</i> Example: <pre>Device(config)# ethernet cfm cc enable level 0-7 evc evc_100</pre>	Globally enables transmission of CCMs.
Step 32	ethernet cfm cc level any evc <i>evc-name</i> interval <i>seconds</i> loss-threshold <i>num-msgs</i> Example: <pre>Device(config)# ethernet cfm cc level any evc evc_100 interval 100 loss-threshold 2</pre>	Sets the parameters for CCMs.

	Command or Action	Purpose
Step 33	end Example: Device(config)# end	Returns to privileged EXEC mode.

What to Do Next



Note

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

Troubleshooting Tips

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

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

Configuring Ethernet OAM Interaction with CFM

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

Configuring the OAM Manager



Note

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ethernet cfm domain <i>domain-name</i> level <i>level-id</i> [direction outward] Example: Device(config)# ethernet cfm domain cstmrl level 3	Defines a CFM domain, sets the domain level, and enters Ethernet CFM configuration mode.
Step 4	service <i>csi-id</i> vlan <i>vlan-id</i> Example: Device(config-ether-cfm)# service csi2 vlan 10	Defines a universally unique customer service instance (CSI) and VLAN ID within the maintenance domain.
Step 5	exit Example: Device(config-ether-cfm)# exit	Returns to global configuration mode.

	Command or Action	Purpose
Step 6	ethernet evc <i>evc-id</i> Example: Device(config)# ethernet evc 50	Defines an EVC and enters EVC configuration mode.
Step 7	oam protocol { cfm svlan <i>svlan-id</i> domain <i>domain-name</i> ldp } Example: Device(config-enc)# oam protocol cfm svlan 10 domain cstmrl	Configures the EVC OAM protocol.
Step 8	exit Example: Device(config-enc)# exit	Returns to global configuration mode.
Step 9	Repeat Steps 3 through 8 to define other CFM domains that you want OAM manager to monitor.	—
Step 10	end Example: Device(config)# end	Returns to privileged EXEC mode.

Enabling Ethernet OAM

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

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

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Device(config)# interface ethernet 1/3	Specifies an interface and enters interface configuration mode.
Step 4	ethernet oam [max-rate <i>oampdus</i> min-rate <i>num-seconds</i>] mode { active passive } timeout <i>seconds</i> Example: Device(config-if)# ethernet oam max-rate 50	Enables Ethernet OAM on an interface.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Configuration Examples for Configuring Ethernet CFM in a Service Provider Network

Example: Provisioning a Network

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

```
CE-A
!
ethernet cfm domain Customer level 7
```

```

!!
ethernet cfm global
ethernet cfm traceroute cache
ethernet cfm traceroute cache size 200
ethernet cfm traceroute cache hold-time 60
!!
ethernet cfm cc level any vlan any interval 20 loss-threshold 3
!
snmp-server enable traps ethernet cfm cc mep-up mep-down cross-connect loop config
snmp-server enable traps ethernet cfm crosscheck mep-missing mep-unknown service-up

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

```

Example: Provisioning Service

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

```

CE-A
!
ethernet cfm domain Customer level 7
service Customer1 vlan 100

```

Example: Provisioning Service

```

!
ethernet cfm global
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 global
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 global
!
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 global
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 global
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 global
!
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 global
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
service Customer1 vlan 100
!
ethernet cfm global
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

```

Glossary

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

