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.

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 is supported *only* on EFP BD with no support on MPLS or Xconnect or VRF.
- CFM is not supported over trunk interface.
- Maintenance endpoints (MEP) statistics for hardware offloaded session do not work.
- We cannot have Port-MEP and MEP over untagged EFP at the same time on the same interface. This is true for default EFP as well, if CFM encapsulation command is not used.
- Hardware offloaded continuity check messages (CCM) intervals are not accurately displayed in the CFM database.
- For Port-MEP, untagged EFP is mandatory. It should be configured for directily connected interface.
- On a port-channel interface with untagged EFP configured, default CFM encapsulation configuration is not recommended.
- Sequence number for hardware offload session is always zero.
- UP MEP hardware session is supported from 16.9.1 release.
- UP MEP hardware CFM packets will be classified under qos-group 0 in egress policy.
- Double tag EFP without rewrite is not supported.
- Port-MEP cannot be configured under port-channel member interface.
- Error counters and output drops are seen in up MEP configured interface.
- MAC address entry for down MEP is not shown in the **show mac-address** table.
- MEP and MIP should not be configured under the same EFP.
- It is not recommended to configure MIPs for hardware offloaded CFM sessions.
- Maximum number of CFM sessions supported system wide is 300.
- Maximum number of CFM sessions per 1G interface is 40.
- Maximum number of CFM sessions per 10G interface is 300.
- CFM UP MEP session is not supported when access and core is configured as DOT1AD NNI.
- CFM UP MEP session is not supported when core is Dot1ad NNI and access as UNI-C.
- Both Software and Hardware CFM session should not be configured under the same EFP.
- Port-MEP and MEP over untagged EFP cannot be configured on the same interface at a time. This is true for default EFP as well, provided CFM encapsulation command is not used.
- Port MEP session should be configured for directly connected interface.
- CFM MIP level dynamic modification is not supported, you need to remove and add new MIP level.

- If UP MEP CFM session configured on a physically down interface, RMEP will not be learnt till the interface comes up.
- CFM over encapsulation priority tagged is not supported.

CFM Configuration over EFP Interface

Ethernet Connectivity Fault Management (CFM) is an end-to-end per-service-instance Ethernet layer OAM protocol that includes proactive connectivity monitoring, fault verification, and fault isolation. Currently, Ethernet CFM supports Up facing and Down facing Maintenance Endpoints (MEPs).

Information About Configuring Ethernet CFM in a Service Provider Network

Ethernet CFM

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

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

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

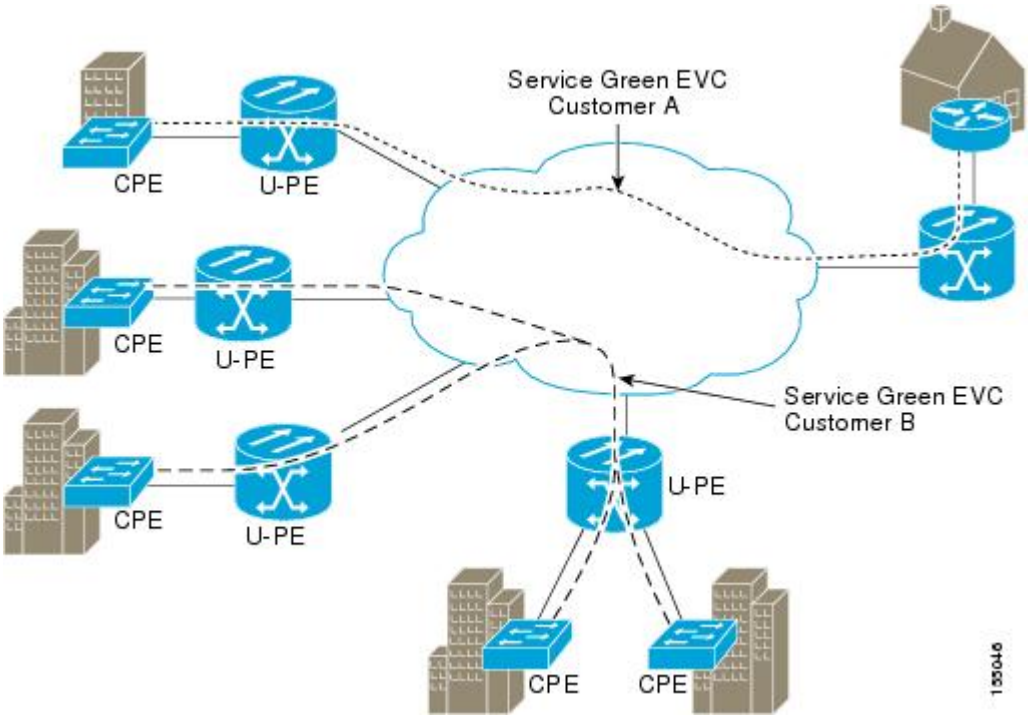
Benefits of Ethernet CFM

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

Customer Service Instance

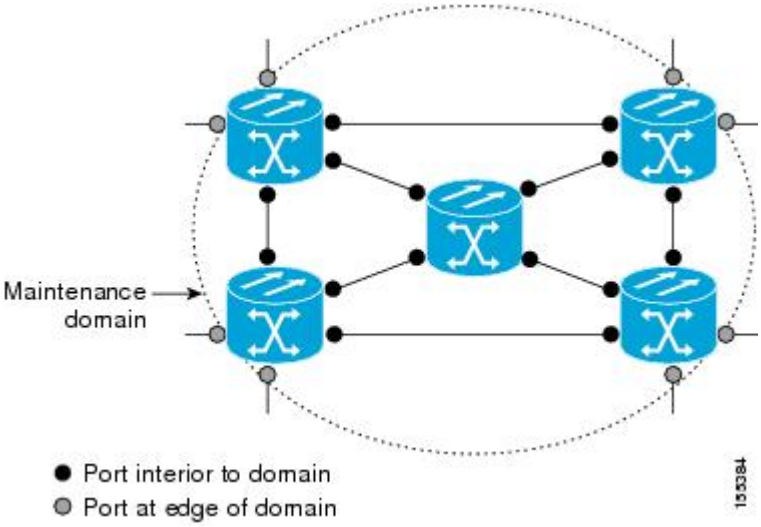
A customer service instance is an Ethernet virtual connection (EVC), which is identified by an S-VLAN within an Ethernet island, and is identified by a globally unique service ID. A customer service instance can be

point-to-point or multipoint-to-multipoint. The figure below shows two customer service instances. Service Instance Green is point to point; Service Instance Blue is multipoint to multipoint.



Maintenance Domain

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



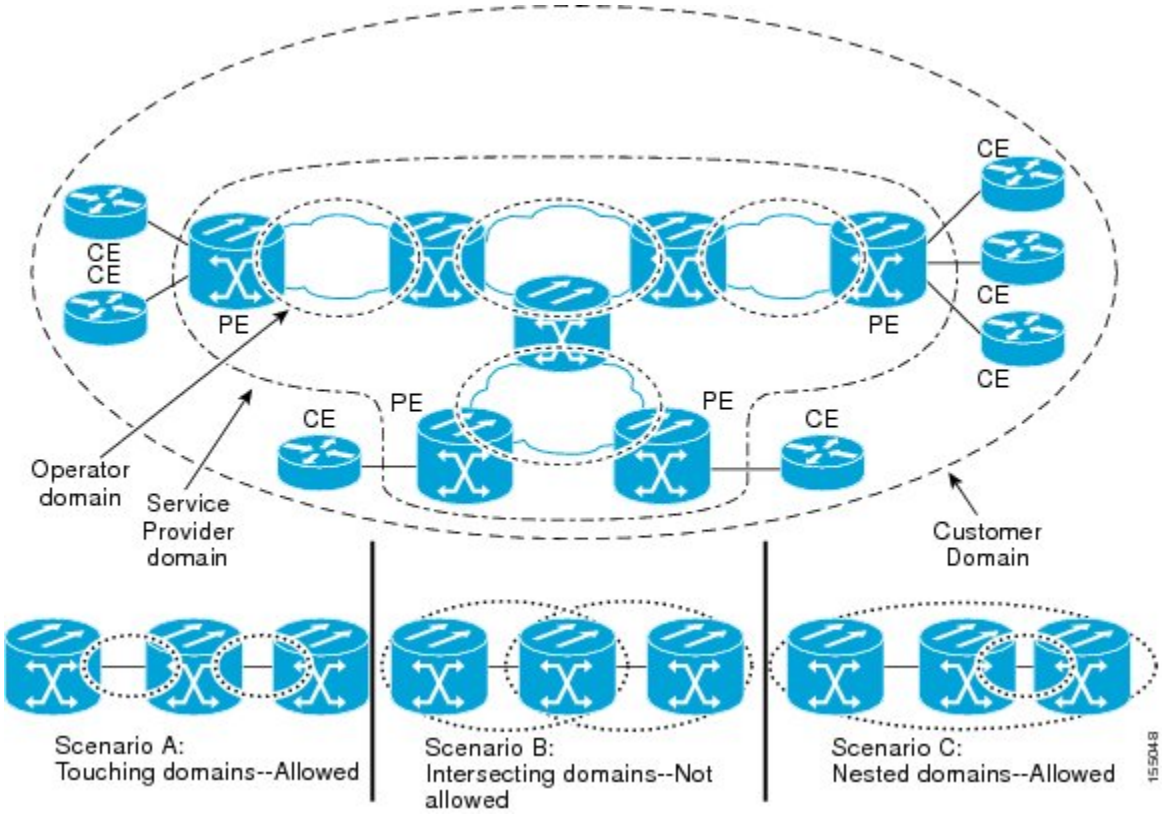
A unique maintenance level in the range of 0 to 7 is assigned to each domain by a network administrator. Levels and domain names are useful for defining the hierarchical relationship that exists among domains. The

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

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

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

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



Maintenance Associations and Maintenance Points

A maintenance association (MA) identifies a service that can be uniquely identified within the maintenance domain. The CFM protocol runs within a maintenance association. A maintenance point is a demarcation

point on an interface that participates in CFM within a maintenance domain. Maintenance points drop all lower-level frames and forward all higher-level frames. There are two types of maintenance points:

- Maintenance end points (MEPs) are points at the edge of the domain that define the boundaries and confine CFM messages within these boundaries. Outward facing or Down MEPs communicate through the wire side (connected to the port). Inward facing or Up MEPs communicate through the relay function side, not the wire side.

  CFM 802.1ag supports up and down per-VLAN MEPs, as well as port MEPs, which are untagged down MEPs that are not associated with a VLAN.

  Port MEPs are configured to protect a single hop and used to monitor link state through CFM. If a port MEP is not receiving continuity check messages from its peer (static remote MEP), for a specified interval, the port is put into an operational down state in which only CFM and OAM packets pass through, and all other data and control packets are dropped.

  - **Up MEP**—An up MEP sends and receives CFM frames through the relay function. It drops all CFM frames at its level or lower that come from the wire side, except traffic going to the down MEP. For CFM frames from the relay side, it processes the frames at its level and drops frames at a lower level. The MEP transparently forwards all CFM frames at a higher level, regardless of whether they are received from the relay or wire side. If the port on which MEP is configured is blocked by STP, the MEP cannot send or receive CFM messages through the relay function. CFM runs at the provider maintenance level (UPE-to-UPE), specifically with up MEPs at the user network interface (UNI).
  - **Down MEP**—A down MEP sends and receives CFM frames through the wire connected to the port on which the MEP is configured. It drops all CFM frames at its level or lower that come from the relay side. For CFM frames from the wire side, it processes all CFM frames at its level and drops CFM frames at lower levels except traffic going to the other lower-level down MEP. The MEP transparently forwards all CFM frames at a higher level, regardless of whether they are received from the relay or through the wire.

- Maintenance intermediate points (MIPs) are internal to a domain, not at the boundary, and respond to CFM only when triggered by traceroute and loopback messages. They forward CFM frames received from MEPs and other MIPs, drop all CFM frames at a lower level (if MIP filtering is enabled), and forward all CFM frames at a higher level and at a lower level and regardless of whether they are received from the relay or wire side. When MIP filtering is enabled, the MIP drops CFM frames at a lower level. MIPs also catalog and forward continuity check messages (CCMs), but do not respond to them.

  MIP filtering is disabled by default, and you can configure it to be enabled or disabled. When MIP filtering is disabled, all CFM frames are forwarded.

  You can manually configure a MIP or configure the device to automatically create a MIP. You can configure a MEP without a MIP. In case of a configuration conflict, manually created MIPs take precedence over automatically created MIPs.

Note MIP filtering and MIP auto-create is not supported.

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 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

Outward facing means that the MEP communicates through the wire.

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

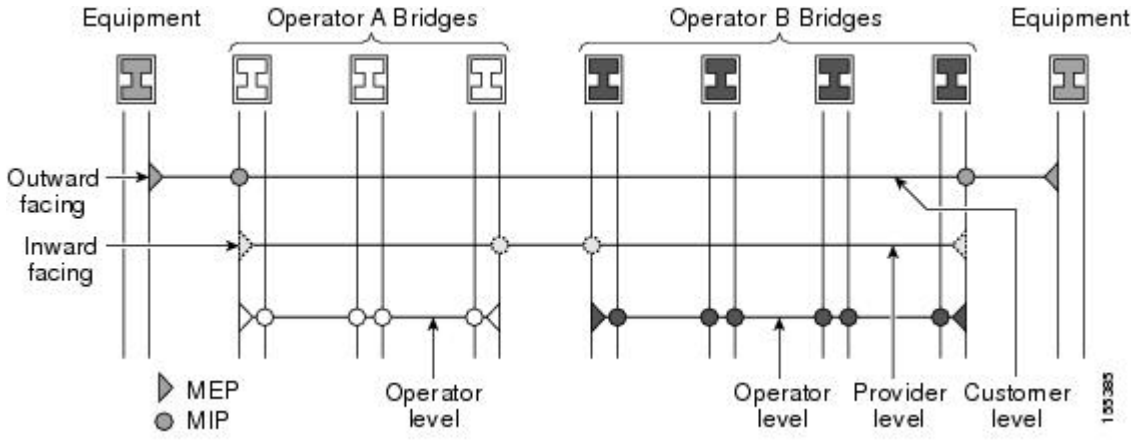
MIPs have the following characteristics:

- 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.
- MIPs 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 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.

Table 1: Feature History Table

Feature Name	Release Information	Description
Unicast MAC for CCM Messages	Cisco IOS XE Bengaluru 17.6.1	Continuity Check Messages (CCM) use multicast destination MAC address by default. This feature enables you to unicast CCM messages to a specific remote MEP (RMEP) to avoid unnecessary traffic flood on the VLAN.

Effective Cisco IOS XE Bengaluru 17.6.1, you can override the multicast function and enable this feature to unicast CCM to destination Remote MEP.

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 10.
- Contains a configurable hold-time value to indicate to the receiver the validity of the message. The default is 3.5 times the transmit interval.
- Catalogued by MIPs at the same maintenance level.
- Terminated by remote MEPs at the same maintenance level.
- Unidirectional and do not solicit a response.
- Carry the status of the port on which the MEP is configured.

Restrictions for Unicast MAC for CCM

- We recommend using the interface MAC address of the destination node for configuring MEP.
- Configure the correct destination MAC address to ensure reachability of unicast CCM.

- NCS 520 series Ethernet Access Devices nodes cannot interoperate with other platforms running on multicast mode.

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.

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.

Both up MEP, down MEP and MIP are supported. If an up MEP is configured under an EFP within a bridge domain, CFM messages would be routed into the bridge, and the rest members of the same bridge domain would be able to receive messages from this MEP. If a down MEP is configured, the messages will not goes into the bridge domain.

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.

Procedure

Step 1 Determine operator level MIPs.

Follow these steps:

- Starting at lowest operator level domain, assign a MIP at every interface internal to the operator network to be visible to CFM.
- Proceed to next higher operator level and assign MIPs.
- Verify that every port that has a MIP at a lower level does not have maintenance points at a higher level.
- Repeat steps a through d until all operator MIPs are determined.

Step 2 Determine operator level MEPs.

Follow these steps:

- Starting at the lowest operator level domain, assign a MEP at every UNI that is part of a service instance.
- Assign a MEP at the network to network interface (NNI) between operators, if there is more than one operator.
- Proceed to next higher operator level and assign MEPs.
- A port with a MIP at a lower level cannot have maintenance points at a higher level. A port with a MEP at a lower level should have either a MIP or MEP at a higher level.

Step 3 Determine service provider MIPs.

Follow these steps:

- Starting at the lowest service provider level domain, assign service provider MIPs at the NNI between operators (if more than one).
- Proceed to next higher service provider level and assign MIPs.
- A port with a MIP at a lower level cannot have maintenance points at a higher level. A port with a MEP at a lower level should not have either a MIP or a MEP at a higher level.

Step 4 Determine service provider MEPs.

Follow these steps:

- Starting at the lowest service provider level domain, assign a MEP at every UNI that is part of a service instance.
- Proceed to next higher service provider level and assign MEPs.
- A port with a MIP at a lower level cannot have maintenance points at a higher level. A port with a MEP at a lower level should have either a MIP or a MEP at a higher level.

Step 5 Determine customer MIPs.

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

- Configure a MIP on every uPE, at the UNI port, in the customer maintenance domain.
- 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.

Configuring Ethernet CFM

Configuring Ethernet CFM consists of the following tasks:

Configuring CFM

Procedure

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **configure terminal**

Example:

```
Device# configure terminal
```

Enters global configuration mode.

Step 3 **ethernet cfm domain** *domain-name* **level** *level-id*

Example:

```
Device(config)# ethernet cfm domain Customer level 7
```

Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.

Step 4 **service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down**

Example:

```
Device(config-ecfm)# service s41 evc 41 vlan 41 direction down
```

Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode.

Note The **direction down** is used only for Down or Outward-facing MEPs. For Up MEPs or Inward-facing MEPs, do not specify **direction down**.

Step 5 **continuity-check**

Example:

```
Device(config-ecfm-srv)# continuity-check
```

Enables 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 time period between CCMs transmission. The default interval is 10 seconds.

Step 7 **exit**

Example:

```
Device(config-ecfm-srv)# exit
```

Returns to Ethernet connectivity fault management configuration mode.

Step 8 **mep archive-hold-time** *minutes*

Example:

```
Device(config-ecfm)# mep archive-hold-time 60
```

Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.

Step 9 **exit**

Example:

```
Device(config-ecfm)# exit
```

Returns to global configuration mode.

Step 10 **ethernet cfm global**

Example:

```
Device(config)# ethernet cfm global
```

Enables CFM processing globally on the device.

Step 11 **etheret cfm ieee**

Example:

```
Router(config)# ethernef cfm ieee
```

Enables CFM IEEE version of CFM.

This command is automatically issued when the ethernet cfm global command is issued.

Step 12 **ethernet cfm traceroute cache**

Example:

```
Device(config)# ethernet cfm traceroute cache
```

Enables caching of CFM data learned through traceroute messages.

Step 13 **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 14 **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 15 **snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**]

Example:

```
Device(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop
cross-connect
```

Enables SNMP trap generation for Ethernet CFM continuity check events.

Step 16 **snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown** | **mep-missing** | **service-up**]

Example:

```
Device(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing
service-up
```

Enables SNMP trap generation for Ethernet CFM continuity check events in relation to the cross-check operation between statically configured MEPS and those learned via CCMs.

Step 17 **end**

Example:

```
Device(config)# end
```

Returns to privileged EXEC mode.

Step 18 **interface** *type number*

Example:

```
Device(config)# interface gigabitethernet0/0/1
```

Specifies an interface and enters interface configuration mode.

Step 19 **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 20 **encapsulation** *encapsulation-type*

Example:

```
Device(config-if-srv)# encapsulation dot1q 5
```

Sets the encapsulation method used by the interface.

Step 21 **bridge-domain** *bridge-id*

Example:

```
Device(config-if-srv)# bridge-domain 100
```

Binds a service instance to a bridge domain instance.

Step 22 **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 23 **end**

Example:

```
Device(config-if-srv)# end
```

Returns to privileged EXEC mode.

Configuring Unicast MAC for CCM

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ethernet cfm domain** *domain-name* **level** *level-id* **Example:** `Device(config)# ethernet cfm domain Customer level 7`	Defines a CFM maintenance domain at a particular maintenance level and enters Ethernet CFM configuration mode.
Step 4	**service** *short-ma-name* **evc** *evc-name* **vlan** *vlanid* **direction down** **Example:** `Device(config-ecfm)# service s41 evc 41 vlan 41 direction down`	Configures a maintenance association within a maintenance domain and enters Ethernet connectivity fault management (CFM) service configuration mode. **Note** The **direction down** is used only for Down or Outward-facing MEPs. For Up MEPs or Inward-facing MEPs, do not specify **direction down**.

	Command or Action	Purpose
Step 5	**continuity-check** **Example:** `Device(config-ecfm-srv)# continuity-check`	Enables 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 time period between CCMs transmission. The default interval is 10 seconds.
Step 7	**exit** **Example:** `Device(config-ecfm-srv)# exit`	Returns to Ethernet connectivity fault management configuration mode.
Step 8	**mep archive-hold-time** *minutes* **Example:** `Device(config-ecfm)# mep archive-hold-time 60`	Sets the amount of time that data from a missing MEP is kept in the continuity check database or that entries are held in the error database before they are purged.
Step 9	**exit** **Example:** `Device(config-ecfm)# exit`	Returns to global configuration mode.
Step 10	**ethernet cfm global** **Example:** `Device(config)# ethernet cfm global`	Enables CFM processing globally on the device.
Step 11	**etheret cfm ieee** **Example:** `Router(config)# ethernef cfm ieee`	Enables CFM IEEE version of CFM. This command is automatically issued when the ethernet cfm global command is issued.
Step 12	**ethernet cfm traceroute cache** **Example:** `Device(config)# ethernet cfm traceroute cache`	Enables caching of CFM data learned through traceroute messages.
Step 13	**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 14	**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.

	Command or Action	Purpose
Step 15	**snmp-server enable traps ethernet cfm cc** [**mep-up**] [**mep-down**] [**config**] [**loop**] [**cross-connect**] **Example:** `Device(config)# snmp-server enable traps ethernet cfm cc mep-up mep-down config loop cross-connect`	Enables SNMP trap generation for Ethernet CFM continuity check events.
Step 16	**snmp-server enable traps ethernet cfm crosscheck** [**mep-unknown** | **mep-missing** | **service-up**] **Example:** `Device(config)# snmp-server enable traps ethernet cfm crosscheck mep-unknown mep-missing service-up`	Enables SNMP trap generation for Ethernet CFM continuity check events in relation to the cross-check operation between statically configured MEPS and those learned via CCMs.
Step 17	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 18	**interface** *type number* **Example:** `Device(config)# interface gigabitethernet0/0/1`	Specifies an interface and enters interface configuration mode.
Step 19	**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 20	**encapsulation** *encapsulation-type* **Example:** `Device(config-if-srv)# encapsulation dot1q 5`	Sets the encapsulation method used by the interface.
Step 21	**bridge-domain** *bridge-id* **Example:** `Device(config-if-srv)# bridge-domain 100`	Binds a service instance to a bridge domain instance.
Step 22	**cfm mep domain** *domain-name* **mpid** *id* **Example:** `Device(config-if-srv)#  cfm mep domain cust1 mpid 1 unicast 00f6.6321.6d95`	Configures the MEP in unicast mode.

	Command or Action	Purpose
Step 23	**end** **Example:** `Device(config-if-srv)# end`	Returns to privileged EXEC mode.

Verifying Unicast MAC for CCM

You can use the **show ethernet cfm maintenance-points local detailed** command to get detailed information on Unicast MAC for CCM on the Gigabit Ethernet interface.

```
Router(config)#show ethernet cfm maintenance-points local detail
Local MEPs:
----------
MPID: 1
DomainName: cust1
Domain ID: cust1
MA Name: s1
Level: 7
Direction: Up
EVC: evc1
Bridge Domain: 10
Service Instance: 10
Interface: Te0/0/22
CC Offload: No
CC-Status: Enabled
CC Loss Threshold: 3
MAC: 00a7.42d1.5ebf
CC Transmission Mode: Unicast
CC Unicast Triggered  Via: Static
CC Unicast Remote Mep Mac Address: 00f6.6321.6d95
LCK-Status: Enabled
LCK Period: 60000(ms)
LCK Expiry Threshold: 3.5
Level to transmit LCK: Default
Defect Condition: No Defect
presentRDI: FALSE
AIS-Status: Enabled
AIS Period: 60000(ms)
AIS Expiry Threshold: 3.5
Level to transmit AIS: Default
Suppress Alarm configuration: Enabled
Suppressing Alarms: No
Source: Static
```

You can use the **show ethernet cfm maintenance-points local** command to view information on Unicast MAC for CCM on the Gigabit Ethernet interface.

```
Router#show ethernet cfm maintenance-points local
Local MEPs:
-------------------------------------------------------------------------------
MPID Domain Name                                Lvl   MacAddress     Type  CC
Ofld Domain Id                                  Dir   Port           Id
     MA Name                                          SrvcInst       Source
     EVC name
     CCM Mode
-------------------------------------------------------------------------------
1    cust1                                      7     00a7.42d1.5ebf BD-V  Y
No   cust1                                      Up    Te0/0/22       10
     s1                                               10             Static
```

```
     evc1
     Unicast
3    sp1                                          5     00a7.42d1.5e82 BD-V  Y
Yes  sp1                                          Down  Gi0/0/2        10
     sa1                                                10             Static
     evc1
     Unicast

Total Local MEPs: 2

Local MIPs: None
```

You can use the **show ethernet cfm maintenance-points remote** command to view information on Unicast MAC for CCM on the Gigabit Ethernet interface.

```
Router(config)#show ethernet cfm maintenance-points remote
-------------------------------------------------------------------------------
MPID  Domain Name                                MacAddress          IfSt  PtSt
 Lvl  Domain ID                                  Ingress
 RDI  MA Name                                    Type Id             SrvcInst
      EVC Name                                                       Age
      Local MEP Info
-------------------------------------------------------------------------------
2     cust1                                      00f6.6321.6dce      Up    Up
 7    cust1                                      Gi0/0/2
 -    s1                                         BD-V 10             10
      evc1                                                           0s
      MPID: 1 Domain: cust1 MA: s1
4     sp1                                        00f6.6321.6d90      Up    Up
 5    sp1                                        Gi0/0/2
 -    sa1                                        BD-V 10             10
      evc1                                                           3s
      MPID: 3 Domain: sp1 MA: sa1

Total Remote MEPs: 2
```

CFM Use Cases

Example For Configuring CFM over Bridge Domain

```
ethernet cfm ieee
ethernet cfm global
ethernet cfm domain cust1 level 7
  service s1 evc 1 vlan 1
   continuity-check
   continuity-check interval 3.3ms

service instance 1 ethernet 1
  encapsulation dot1q 1
  bridge-domain 1
  cfm mep domain cust1 mpid 1
```

Example For Configuring CFM over Default Encapsulation

```
ethernet cfm domain oper2 level 7
service cust1 evc 1000 vlan 1500 direction down
  continuity-check
  continuity-check interval 3.3ms

service instance 1000 ethernet 1000
  encapsulation default
  bridge-domain 1500
```

```
cfm mep domain cust1 mpid 8191
cfm encapsulation dot1q 1500
```

Verification Commands for CFM

Use the following commands to verify CFM:

- **show ethernet cfm maintenance-points local**
- **show ethernet cfm maintenance-points remote**
- **show ethernet cfm statistics**
- **show ethernet cfm ccm-learning-database**
- **show ethernet cfm errors**

SNMP Traps

The support provided by the Cisco IOS XE software implementation of Ethernet 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.

Steps to Generate SNMP Traps for CFM

To generate SNMP traps, following commands need to be configured on the router.

```
ethernet cfm logging
logging snmp-trap 0 7
logging history debugging
```

Send Trap to SNMP Server

```
snmp-server enable traps ethernet cfm cc [mep-up] [mep-down] [config] [loop] [cross-connect]
snmp-server enable traps ethernet cfm crosscheck [mep-unknown] [mep-missing] [ service-up]
```

Note If syslog trap is enabled, by default trap is generated for messages of severity level emergency, alert, critical, error and warning (0-4). For other severity levels need to enable **logging snmp-trap 0 7** and **logging history debugging**

```
Router(config)#ethernet cfm logging
Router(config)#logging snmp-trap 0 7
Router(config)#logging history debugging
Router(config)#snmp-server enable traps ethernet cfm cc
Router(config)#snmp-server enable traps ethernet cfm crosscheck
```

Logs for MEP going DOWN

```
Console-logs:

Router(config)#
*Oct 26 21:32:06.663 IST: %E_CFM-3-REMOTE_MEP_DOWN: Remote MEP mpid 10 evc 2 vlan 2 MA name
 s2 in domain cust2 changed state to down with event code TimeOut.
*Oct 26 21:32:06.664 IST: %E_CFM-6-ENTER_AIS: local mep with mpid 20 level 2 BD/VLAN 2 dir
 D Interface Te0/3/1 enters AIS defect condition
*Oct 26 21:32:09.147 IST: %E_CFM-3-FAULT_ALARM: A fault has occurred in the network for the
 local MEP having mpid 20 evc 2 vlan 2 for service MA name s2 with the event code
DefRemoteCCM.
```

SNMP Server Side Logs

Received SNMPv2c Trap

```
Community: public
From: 7.32.22.154
sysUpTimeInstance = 04:00:54.27
snmpTrapOID.0 = clogMessageGenerated
clogHistFacility.76 = E_CFM
clogHistSeverity.76 = error(4)
clogHistMsgName.76 = REMOTE_MEP_DOWN
clogHistMsgText.76 = Remote MEP mpid 10 evc 2 vlan 2 MA name s2 in domain cust2 changed
state to down with event code TimeOut.
clogHistTimestamp.76 = 04:00:54.27
```

Received SNMPv2c Trap

```
Community: public
From: 7.32.22.154
sysUpTimeInstance = 04:00:54.27
snmpTrapOID.0 = clogMessageGenerated
clogHistFacility.77 = E_CFM
clogHistSeverity.77 = info(7)
clogHistMsgName.77 = ENTER_AIS
clogHistMsgText.77 = local mep with mpid 20 level 2 BD/VLAN 2 dir D Interface Te0/3/1 enters
 AIS defect condition
clogHistTimestamp.77 = 04:00:54.27
```

Received SNMPv2c Trap

```
Community: public
From: 7.32.22.154
sysUpTimeInstance = 04:00:56.75
snmpTrapOID.0 = dot1agCfmFaultAlarm
dot1agCfmMepHighestPrDefect.10.2.20 = defRemoteCCM(3)
```

Received SNMPv2c Trap

```
Community: public
From: 7.32.22.154
sysUpTimeInstance = 04:00:56.75
snmpTrapOID.0 = clogMessageGenerated
clogHistFacility.78 = E_CFM
clogHistSeverity.78 = error(4)
clogHistMsgName.78 = FAULT_ALARM
clogHistMsgText.78 = A fault has occurred in the network for the local MEP having mpid 20
evc 2 vlan 2 for service MA name s2 with the event code DefRemoteCCM.
clogHistTimestamp.78 = 04:00:56.75
```

Logs for MEP Coming Up

Console-logs

```
================================================
Router(config)#
*Oct 26 21:35:03.780 IST: %E_CFM-6-REMOTE_MEP_UP: Continuity Check message is received from
 a remote MEP with mpid 10 evc 2 vlan 2 MA name s2 domain cust2 interface status Up event
code Returning.
*Oct 26 21:35:03.781 IST: %E_CFM-6-EXIT_AIS: local mep with mpid 20 level 2 BD/VLAN 2 dir
D Interface Te0/3/1 exited AIS defect condition
```

SNMP Server Side Logs

Received SNMPv2c Trap

```
================================================
Community: public
From: 7.32.22.154
sysUpTimeInstance = 04:03:51.39
snmpTrapOID.0 = clogMessageGenerated
clogHistFacility.79 = E_CFM
clogHistSeverity.79 = info(7)
clogHistMsgName.79 = REMOTE_MEP_UP
clogHistMsgText.79 = Continuity Check message is received from a remote MEP with mpid 10
evc 2 vlan 2 MA name s2 domain cust2 interface status Up event code Returning.
clogHistTimestamp.79 = 04:03:51.38
```

Received SNMPv2c Trap

```
Community: public
From: 7.32.22.154
```

```
sysUpTimeInstance = 04:03:51.39
snmpTrapOID.0 = clogMessageGenerated
clogHistFacility.80 = E_CFM
clogHistSeverity.80 = info(7)
clogHistMsgName.80 = EXIT_AIS
clogHistMsgText.80 = local mep with mpid 20 level 2 BD/VLAN 2 dir D Interface Te0/3/1 exited
 AIS defect condition
clogHistTimestamp.80 = 04:03:51.38
```

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.

Troubleshooting CFM Features

Provides troubleshooting solutions for the CFM features.

Table 2: Troubleshooting Scenarios for CFM Features

Problem	Solution
When you configure CFM, the message “Match registers are not available” is displayed.	For more information on match registers, see Ethernet Co Fault Management at http://www.cisco.com/en/US/docs/ios/12_2sr/12_2sra/feature/guide/sr CFM uses two match registers to identify the control pa and each VLAN spanning tree also uses a match register its control packet type. For both protocols to work on t system, each line card should support three match regis at least one supporting only a 44 bit MAC match.
CFM configuration errors	CFM configuration error occurs when when a MEP rec continuity check with an overlapping MPID. To verify of the error, use the command **show ethernet cfm erro configuration** or **show ethernet cfm errors**.

Problem	Solution
CFM ping and traceroute result is "not found"	Complete these steps: **1.** Use **show run | i ethernet cfm** to view all CFM configurations. **2.** Use **show ethernet cfm statistics** to view loca their CCM statistics **3.** Use **trace ethernet cfm** command to start a CF
CFM connectivity is down and issues at the maintenance domain levels	Use the **ping ethernet {mac-address | mpid** *id* **| m** **domain** *domain-name* { **vlan v***lan-id* **| port | evc** *ev* **the traceroute ethernet {mac-address | mpid** *id* **}** *domain-name* { **vlan** *vlan-id* **| port | evc evc-name}** to verify ethernet CFM connectivity. Share the outp for further investigation. **Note** CFM multicast ping with packet size g 1460 is not supported.

Loop trap error	Use the **show ethernet cfm error** command to che Trap errors as shown here:
	`CE(config-if)#do sh ethernet cfm err`
	`Level Vlan MPID Remote MAC Reason Service ID`
	`5 711 550 1001.1001.1001 Loop Trap Er`
	`PE#sh ethernet cfm err`
	`Level Vlan MPID Remote MAC Reason Service ID`
	`5 711 550 1001.1001.1001 Loop Trap Er`

Module has insufficient match registers	Complete these steps: **1.** Verify and confirm if a unsupported line card is the router. **2.** If yes, perform an OIR of the unsupported line
CFM is deactivated	Complete these steps: **1.** Check if all the line cards have free match reag **2.** Check if CFM is activated on supervisor cards. supported on supervisor cards that has two matc this scenario, CFM is automatically disabled on and enabled on the remaining line cards.

Problem	Solution
ethernet cfm logging	In a scale scenario, you configure either the console log rate-limiting using **logging rate-limit** or using **logging** instead of using **logging console**. The suggested rate-limi 30 messages per second.