



Configuring Ethernet OAM

This module describes the configuration of Ethernet Operations, Administration, and Maintenance (OAM):

Table 1: Feature Information Table

Release	Modification
Release 7.3.1	Support for Ethernet Link OAM was introduced.

- [Information About Configuring Ethernet OAM, on page 1](#)
- [Configuration Examples for Ethernet OAM, on page 5](#)
- [Ethernet CFM, on page 7](#)
- [Unidirectional link detection protocol, on page 24](#)
- [How to Configure Ethernet OAM, on page 30](#)
- [CFM Over Bundles, on page 52](#)
- [Ethernet SLA Statistics Measurement in a Profile, on page 54](#)
- [Ethernet frame delay measurement for L2VPN services, on page 58](#)
- [Minimum delay bin, on page 63](#)
- [Link loss forwarding, on page 66](#)

Information About Configuring Ethernet OAM

To configure Ethernet OAM, you should understand the following concepts:

Ethernet Link OAM

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
Ethernet Link OAM support	Release 25.2.1	Introduced in this release on: Fixed Systems (8700 [ASIC:K100]) (select variants only*) This release supports: Ethernet Link OAM on physical interface— (802.3ah) link monitoring. *This feature is now supported on Cisco 8712-MOD-M routers.
Ethernet Link OAM on Physical Interface— (802.3ah) Link Monitoring	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) This feature is now supported on: 8011-4G24Y4H-I
Ethernet Link OAM on Physical Interface— (802.3ah) Link Monitoring	Release 24.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100]) (select variants only*) Ethernet link OAM operates on a single, physical link and it can be configured to monitor either side or both sides of that link. Ethernet OAM supports link monitoring: * This feature is supported on Cisco 8712-MOD-M routers.
Ethernet Link OAM	Release 7.3.1	This feature allows service providers to monitor the quality of the connections on a MAN or WAN. Service providers can monitor specific events, and take actions on events. Ethernet link OAM operates on a single, physical link and it can be configured to monitor either side or both sides of that link.

Ethernet as a Metro Area Network (MAN) or a Wide Area Network (WAN) technology benefits greatly from the implementation of Operations, Administration and Maintenance (OAM) features. Ethernet link OAM features allow Service Providers to monitor the quality of the connections on a MAN or WAN. Service

providers can monitor specific events, and take actions on events. Ethernet link OAM operates on a single, physical link and it can be configured to monitor either side or both sides of that link.

Ethernet link OAM can be configured in the following ways:

- A Link OAM profile can be configured, and this profile can be used to set the parameters for multiple interfaces.
- Link OAM can be configured directly on an interface.

When an interface is also using a link OAM profile, specific parameters that are set in the profile can be overridden by configuring a different value directly on the interface.

An Ethernet Link OAM profile simplifies the process of configuring EOAM features on multiple interfaces. An Ethernet OAM profile, and all of its features, can be referenced by other interfaces, allowing other interfaces to inherit the features of that Ethernet OAM profile.

Individual Ethernet link OAM features can be configured on individual interfaces without being part of a profile. In these cases, the individually configured features always override the features in the profile.

The preferred method of configuring custom EOAM settings is to create an EOAM profile in Ethernet configuration mode and then attach it to an individual interface or to multiple interfaces.

These standard Ethernet Link OAM features are supported on the router:

Neighbor Discovery

Neighbor discovery enables each end of a link to learn the OAM capabilities of the other end and establish an OAM peer relationship. Each end also can require that the peer have certain capabilities before it will establish a session. You can configure certain actions to be taken if there is a capabilities conflict or if a discovery process times out, using the **action capabilities-conflict** or **action discovery-timeout** commands.

EFD

Ethernet Fault Detection (EFD) is a mechanism that allows Ethernet OAM protocols to control the `line protocol` state of an interface.

Unlike many other interface types, Ethernet interfaces do not have a line protocol, whose state is independent from that of the interface. For Ethernet interfaces, this role is handled by the physical-layer Ethernet protocol itself, and therefore if the interface is physically up, then it is available and traffic can flow.

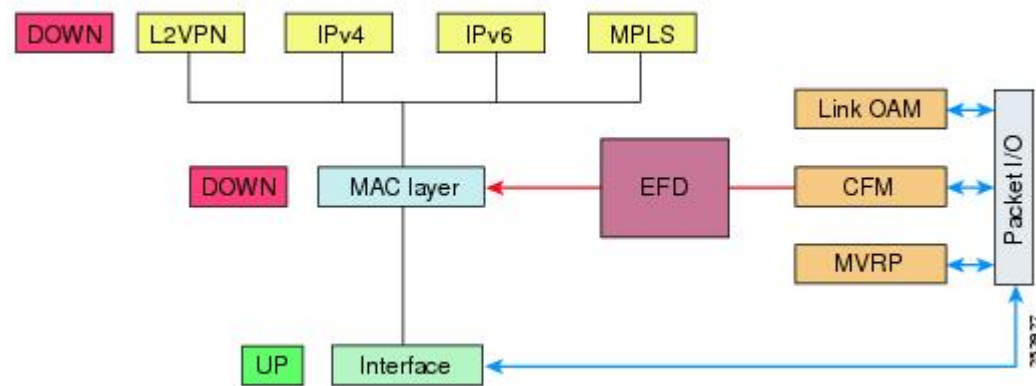
EFD changes this to allow EOAM to act as the line protocol for Ethernet interfaces. This allows EOAM to control the interface state so that if a EOAM defect (such as AIS or loss of continuity) is detected with an expected peer MEP, the interface can be shut down. This not only stops traffic flow, but also triggers actions in any higher-level protocols to route around the problem. For example, in the case of Layer 2 interfaces, the MAC table would be cleared and MSTP would reconverge. For Layer 3 interfaces, the ARP cache would be cleared and potentially the IGP would reconverge.



Note EFD can only be used for down MEPs. When EFD is used to shut down the interface, the EOAM frames continue to flow. This allows EOAM to detect when the problem has been resolved, and thus bring the interface backup automatically.

This figure shows EOAM detection of an error on one of its sessions EFD signaling an error to the corresponding MAC layer for the interface. This triggers the MAC to go to a down state, which further triggers

Figure 1: EOAM Error Detection and EFD Trigger



MIB retrieval enables an OAM peer on one side of an interface to get the MIB variables from the remote side of the link. The MIB variables that are retrieved from the remote OAM peer are READ ONLY.

Miswiring Detection is a Cisco-proprietary feature that uses the 32-bit vendor field in every Information OAMPDU to identify potential miswiring cases.

SNMP traps can be enabled or disabled on an Ethernet OAM interface.

Link monitoring enables an OAM peer to monitor faults that cause the quality of a link to deteriorate over time. When link monitoring is enabled, an OAM peer can be configured to take action when the configured thresholds are exceeded.

Remote loopback enables one side of a link to put the remote side of the link into loopback mode for testing. When remote loopback is enabled, all packets initiated by the primary side of the link are looped back to the primary side, unaltered by the remote side. In remote loopback mode, the remote side is not allowed to inject any data into the packets.

Configuration Examples for Ethernet OAM

This section provides the following configuration examples:

Configuring Ethernet OAM Features on an Individual Interface: Example

This example shows how to configure Ethernet OAM features on an individual interface:

```
configure
interface TenGigE 0/1/0/0
  ethernet oam
  connection timeout 30
  mib-retrieval
  link-monitor
    frame window milliseconds 60000
    frame threshold low 10000000 high 60000000
    frame-period window milliseconds 60000
    frame-period threshold ppm low 100 high 120000
    frame-seconds window milliseconds 900000
    frame-seconds threshold low 3 high 900
    symbol-period window milliseconds 60000
    symbol-period threshold ppm low 1000000 high 1000000
  exit
  require-remote
  mode active
  mib-retrieval
exit
action
  critical-event error-disable-interface
  dying-gasp error-disable-interface
  capabilities-conflict error-disable-interface
  wiring-conflict error-disable-interface
  discovery-timeout error-disable-interface
  session-down error-disable-interface
commit
```

Configuring an Ethernet OAM Profile Globally: Example

This example shows how to configure an Ethernet OAM profile globally:

```
configure
ethernet oam profile Profile_1
  connection timeout 30
  mib-retrieval
  link-monitor
    frame window milliseconds 60000
    frame threshold low 10000000 high 60000000
    frame-period window milliseconds 60000
    frame-period threshold ppm low 100 high 1000000
    frame-seconds window milliseconds 900000
    frame-seconds threshold low 3 high 900
    symbol-period window milliseconds 60000
    symbol-period threshold ppm low 1000000 high 1000000
  exit
  require-remote
  mode active
  mib-retrieval
```

```

exit
action
critical-event error-disable-interface
dying-gasp error-disable-interface
capabilities-conflict error-disable-interface
wiring-conflict error-disable-interface
discovery-timeout error-disable-interface
session-down error-disable-interface
commit

```

Configuring Ethernet OAM Features to Override the Profile on an Individual Interface: Example

This example shows the configuration of Ethernet OAM features in a profile followed by an override of that configuration on an interface:

```

configure
ethernet oam profile Profile_1
mode passive
action dying-gasp disable
action critical-event disable
action discovery-timeout disable
action session-up disable
action session-down disable
action capabilities-conflict disable
action wiring-conflict disable

commit

configure
interface TenGigE 0/1/0/0
ethernet oam
profile Profile_1
mode active
action dying-gasp log
action critical-event log
action discovery-timeout log
action session-up log
action session-down log
action capabilities-conflict log
action wiring-conflict log

commit

```

Recovering from error-disable: Example

You can recover an error-disabled interface due to session-down using one of these methods:

- Manually clear the error-disable using the **clear** command.

```

Router# configure
Router(config)# ethernet oam profile Profile_1
Router(config-eoam)# action
Router(config-eoam-action)# clear session-down error-disable-interface

```

- Disable and then re-enable the network link using administrative shutdown commands to reset the connection.

```

Router# configure
Router(config)# interface TenGigE 0/1/0/0
Router(config-if)# shutdown
Router(config-if)# commit
Router(config-if)# no shutdown
Router(config-if)# commit

```

- Configure an auto-recovery timer for this error-disable reason.

```

Router# configure
Router(config)# error-disable recovery cause link-oam-session-down interval 30
Router(config)# commit

```

Clearing Ethernet OAM Statistics on an Interface: Example

This example shows how to clear Ethernet OAM statistics on an interface:

```
RP/0/RP0/CPU0:router# clear ethernet oam statistics interface gigabitethernet 0/1/5/1
```

Enabling SNMP Server Traps on a Router: Example

This example shows how to enable SNMP server traps on a router:

```

configure
snmp-server traps ethernet oam events

```

Ethernet CFM

Table 3: Feature History Table

Feature name	Release	Description
Y.1731 support on CFM	Release 25.3.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) This feature is now supported on: • 8011-4G24Y4H-I • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E

Feature name	Release	Description
Increase in number of CFM sessions	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100]) This feature is now supported on: • 8712-MOD-M • 8011-4G24Y4H-I
Increase in number of CFM sessions	Release 24.4.1	Introduced in this release on: Fixed Systems(8200, 8700); Centralized Systems (8600); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]) The number of supported Connectivity Fault Management (CFM) sessions is now increased to 500. This enhancement improves fault detection, network visibility, scalability, and troubleshooting, which are crucial for managing high-performance networks.
CFM on bundle member link for connectivity check	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) This feature is now supported on: • 8011-4G24Y4H-I
CFM on bundle member link for connectivity check	Release 24.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100]) (select variants only*) This feature introduces support for Connectivity Fault Management (CFM) on bundle members. * This feature is supported on Cisco 8712-MOD-M routers.
Up MEP and down MEP support in CFM	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) This feature is now supported on: • 8011-4G24Y4H-I

Feature name	Release	Description
Up MEP and down MEP support in CFM	Release 24.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100]) (select variants only*) This feature introduces Maintenance End Points (MEP) entities that you can configure in a domain. * This feature is supported on Cisco 8712-MOD-M routers.
Monitoring Layer 3 connectivity using down MEP on L3 interfaces	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) This feature is now supported on: • 8011-4G24Y4H-I
Monitoring Layer 3 connectivity using down MEP on L3 interfaces	Release 24.4.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100, K100]) (select variants only*); Modular Systems (8800 [LC ASIC: P100]) This enhancement expands network diagnostics to L3 interfaces at L2 network termination, simplifying the management and maintenance of multilayer networks. * This feature is supported on Cisco 8712-MOD-M routers.
Monitoring Layer 3 connectivity using down MEP on L3 interfaces	Release 24.2.11	This enhancement expands network diagnostics to L3 interfaces at L2 network termination, simplifying the management and maintenance of multilayer networks. Without impacting the underlying L2 infrastructure, this feature uses CFM packets to verify the connection of L3 paths. Previously, CFM Down MEP support was limited to L2 interfaces associated with cross-connect or bundle members. This feature is supported on both physical main and subinterfaces, bundle main and subinterfaces.

Feature name	Release	Description
CFM on bundle member link for connectivity check	Release 7.3.15	This feature introduces support for Connectivity Fault Management (CFM) on bundle members. Earlier, network administrators managed networks by using the fault, configuration, account, performance, security model. CFM is one of a suite of the Ethernet OAM protocols, which uses a combination of keepalive packets and MAC-based pings, and traceroutes to detect faults in a network. With the CFM feature, you: • reduce operating expenses for service operators by reducing network faults and errors • provide end-to-end maintenance of networks
Up MEP and down MEP support in CFM	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100]) This feature is now supported on: • 8712-MOD-M • 8011-4G24Y4H-I
Up MEP and down MEP support in CFM	Release 7.3.15	This feature introduces Maintenance End Points (MEP) entities that you can configure in a domain. MEPs send either CFM frames from the interface where they are configured or CFM frames that are received on other interfaces. MEPs allow you to perform fault management and carry out performance checks.

Ethernet Connectivity Fault Management (CFM) is a service-level OAM protocol that provides tools for monitoring and troubleshooting end-to-end Ethernet services per VLAN. This includes proactive connectivity monitoring, fault verification, and fault isolation. CFM uses standard Ethernet frames and can be run on any physical media that is capable of transporting Ethernet service frames. Unlike most other Ethernet protocols

which are restricted to a single physical link, CFM frames can transmit across the entire end-to-end Ethernet network.

CFM is defined in two standards:

- IEEE 802.1ag—Defines the core features of the CFM protocol.
- ITU-T Y.1731—Redefines, but maintains compatibility with the features of IEEE 802.1ag, and defines some additional features.

Ethernet CFM supports these functions of ITU-T Y.1731:

- ETH-CC, ETH-RDI, ETH-LB, ETH-LT—These are equivalent to the corresponding features defined in IEEE 802.1ag.



Note The Linktrace responder procedures defined in IEEE 802.1ag are used rather than the procedures defined in Y.1731; however, these are interoperable.

- ETH-AIS—The reception of ETH-LCK messages is also supported.

Limitations and restrictions

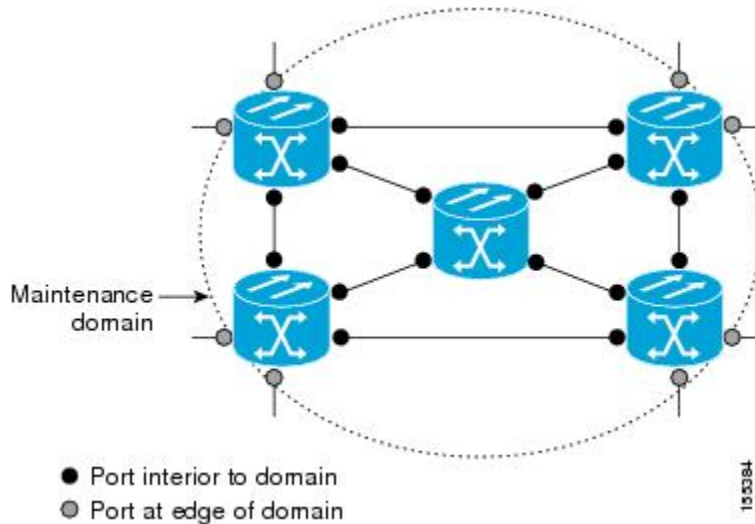
- The system supports only cross-connect.
- MIPs are not supported.
- Supports timer of 1s, 10s, 1m, 10m.
- Supports timer of 100ms, 1s, 10s, 1m, 10m for bundle members.
- Multiple MEPs of different directions are not supported on the same interface or Xconnect.
- CFM is not supported on L2 subinterfaces with default encapsulation.
- When configuring CFM down MEP on an interface, ensure that the interface is included in an L2VPN.

Maintenance Domains

To understand how the CFM maintenance model works, you need to understand these concepts and features:

A maintenance domain describes 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 interfaces internal to it and at its boundary, as shown in this figure.

Figure 2: CFM Maintenance Domain



A maintenance domain is defined by the bridge ports that are provisioned within it. Domains are assigned maintenance levels, in the range of 0 to 7, by the administrator. The level of the domain is useful in defining the hierarchical relationships of multiple domains.

CFM maintenance domains allow different organizations to use CFM in the same network, but independently. For example, consider a service provider who offers a service to a customer, and to provide that service, they use two other operators in segments of the network. In this environment, CFM can be used in the following ways:

- The customer can use CFM between their CE devices, to verify and manage connectivity across the whole network.
- The service provider can use CFM between their PE devices, to verify and manage the services they are providing.
- Each operator can use CFM within their operator network, to verify and manage connectivity within their network.

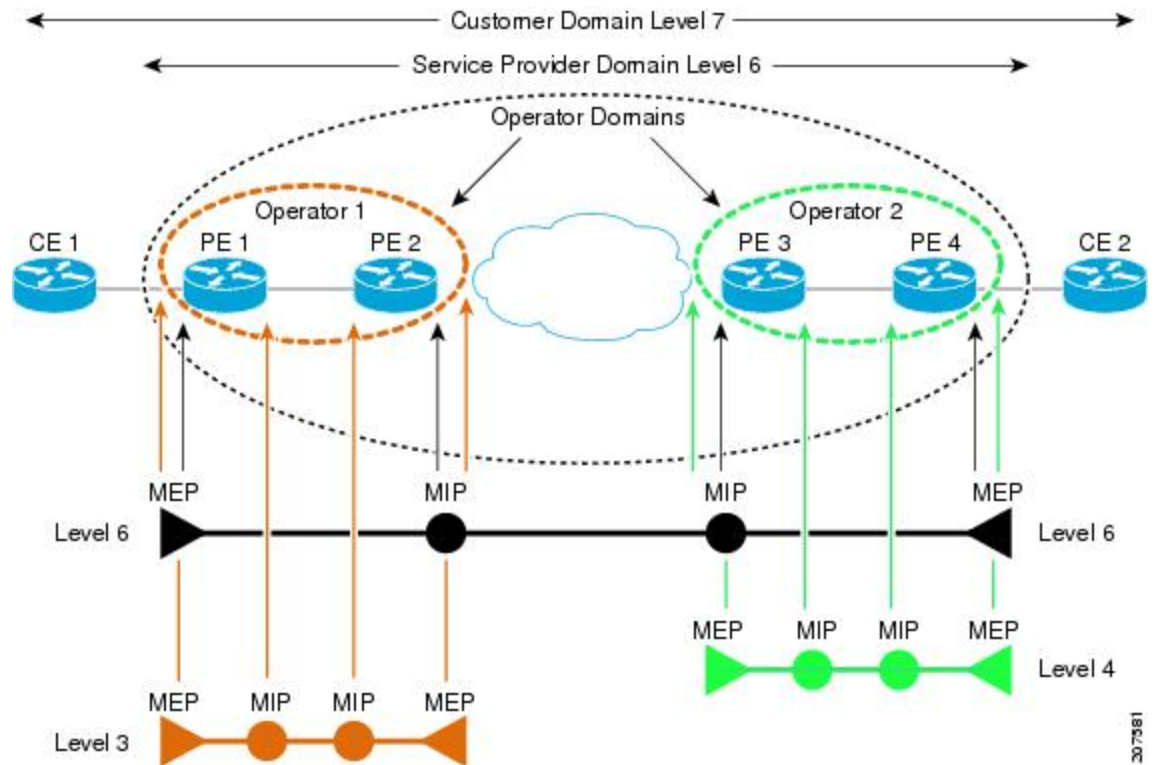
Each organization uses a different CFM maintenance domain.

This figure shows an example of the different levels of maintenance domains in a network.



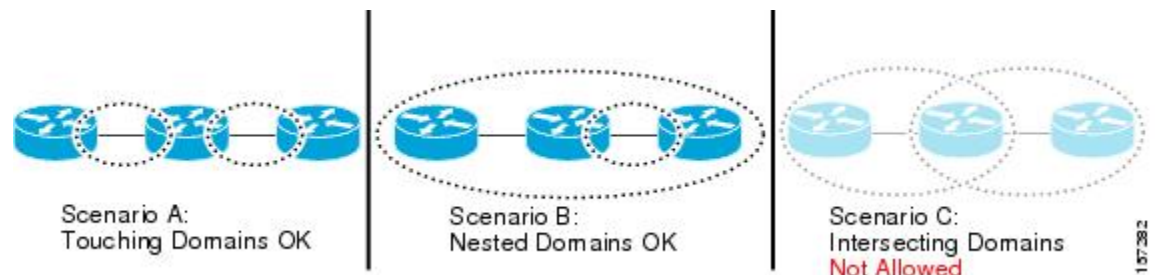
Note In CFM diagrams, the conventions are that triangles represent MEPs, pointing in the direction that the MEP sends CFM frames, and circles represent MIPs.

Figure 3: Different CFM Maintenance Domains Across a Network



To ensure that the CFM frames for each domain do not interfere with each other, each domain is assigned a maintenance level, between 0 and 7. Where domains are nested, as in this example, the encompassing domain must have a higher level than the domain it encloses. In this case, the domain levels must be negotiated between the organizations involved. The maintenance level is carried in all CFM frames that relate to that domain.

CFM maintenance domains may touch or nest, but cannot intersect. This figure illustrates the supported structure for touching and nested domains, and the unsupported intersection of domains.



Services

A CFM service allows an organization to partition its CFM maintenance domain, according to the connectivity within the network. For example, if the network is divided into a number of virtual LANs (VLANs), a CFM service is created for each of these. CFM can then operate independently in each service. It is important that the CFM services match the network topology, so that CFM frames relating to one service cannot be received in a different service. For example, a service provider may use a separate CFM service for each of their customers, to verify and manage connectivity between that customer's end points.

A CFM service is always associated with the maintenance domain that it operates within, and therefore with that domain's maintenance level. All CFM frames relating to the service carry the maintenance level of the corresponding domain.



Note CFM Services are referred to as *Maintenance Associations* in IEEE 802.1ag and as *Maintenance Entity Groups* in ITU-T Y.1731.

Maintenance Points

A CFM Maintenance Point (MP) is an instance of a particular CFM service on a specific interface. CFM only operates on an interface if there is a CFM maintenance point on the interface; otherwise, CFM frames are forwarded transparently through the interface.

A maintenance point is always associated with a particular CFM service, and therefore with a particular maintenance domain at a particular level. Maintenance points generally only process CFM frames at the same level as their associated maintenance domain. Frames at a higher maintenance level are always forwarded transparently, while frames at a lower maintenance level are normally dropped. This helps enforce the maintenance domain hierarchy, and ensures that CFM frames for a particular domain cannot leak out beyond the boundary of the domain.

There are following type(s) of MP(s):

- **Maintenance End Points (MEPs)**—Created at the edge of the domain. Maintenance end points (MEPs) are members of a particular service within a domain and are responsible for sourcing and sinking CFM frames. They periodically transmit continuity check messages and receive similar messages from other MEPs within their domain. They also transmit traceroute and loopback messages at the request of the administrator. MEPs are responsible for confining CFM messages within the domain.

MEP and CFM Processing Overview

The boundary of a domain is an interface, rather than a bridge or host. Therefore, MEPs can be sub-divided into two categories:

- **Down MEPs**—Send CFM frames from the interface where they are configured, and process CFM frames received on that interface. Down MEPs transmit AIS messages upward (toward the cross-connect).
- **Up MEPs**—Send frames into the bridge relay function, as if they had been received on the interface where the MEP is configured. They process CFM frames that have been received on other interfaces, and have been switched through the bridge relay function as if they are going to be sent out of the interface where the MEP is configured. Up MEPs transmit AIS messages downward (toward the wire). However,

AIS packets are only sent when there is a MIP configured on the same interface as the MEP and at the level of the MIP.

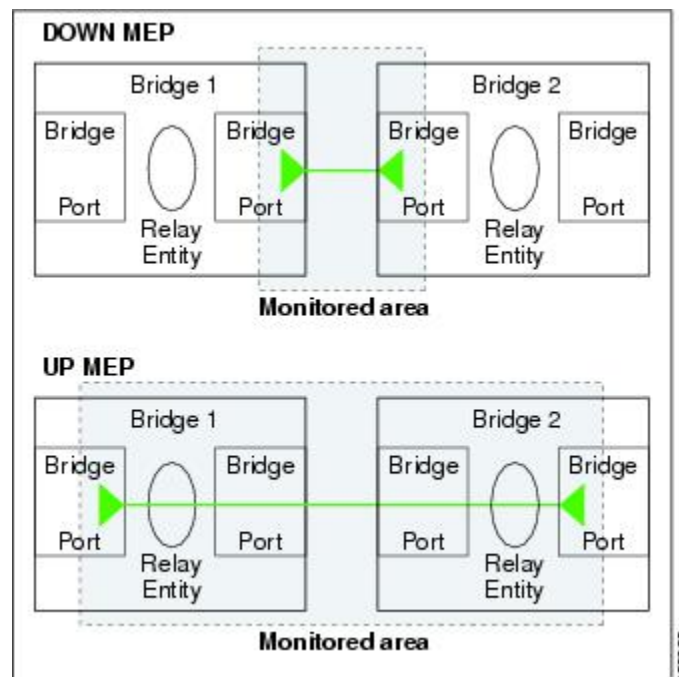


Note

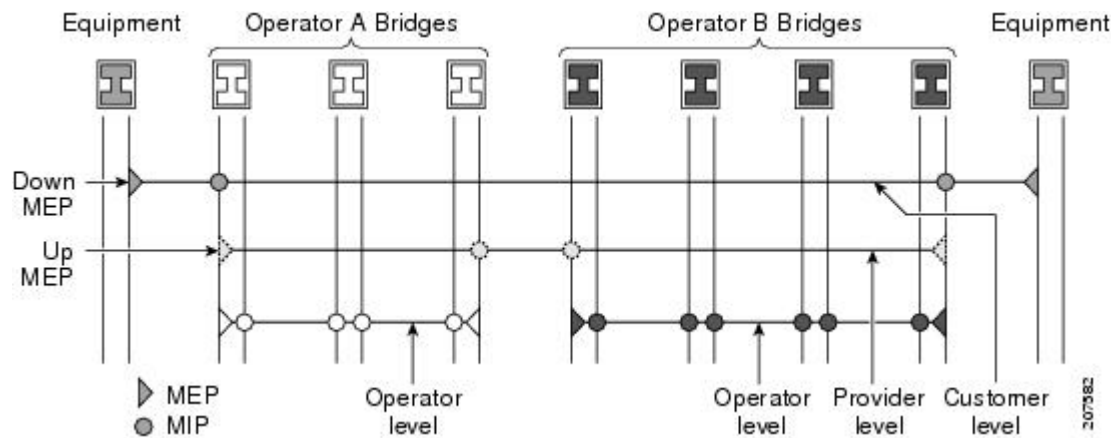
- The terms *Down MEP* and *Up MEP* are defined in the IEEE 802.1ag and ITU-T Y.1731 standards, and refer to the direction that CFM frames are sent from the MEP. The terms should not be confused with the operational status of the MEP.
- The router only supports the “Down MEP level < Up MEP level” configuration.

This figure illustrates the monitored areas for Down and Up MEPs.

Figure 4: Monitored Areas for Down and Up MEPs



This figure shows maintenance points at different levels. Because domains are allowed to nest but not intersect, a MEP at a low level often corresponds with a MEP at a higher level.



Up MEPs can only exist on switched (Layer 2) interfaces, because they send and receive frames from the bridge relay function. Down MEPs can be created on switched (Layer 2) interfaces.

MEPs continue to operate normally if the interface they are created on is blocked by the Spanning Tree Protocol (STP); that is, CFM frames at the level of the MEP continue to be sent and received, according to the direction of the MEP. MEPs never allow CFM frames at the level of the MEP to be forwarded, so the STP block is maintained.



Note

A separate set of CFM maintenance levels is created every time a VLAN tag is pushed onto the frame. Therefore, if CFM frames are received on an interface which pushes an additional tag, so as to “tunnel” the frames over part of the network, the CFM frames will not be processed by any MPs within the tunnel, even if they are at the same level. For example, if a CFM MP is created on an interface with an encapsulation that matches a single VLAN tag, any CFM frames that are received at the interface that have two VLAN tags will be forwarded transparently, regardless of the CFM level.

CFM Protocol Messages

The CFM protocol consists of a number of different message types, with different purposes. All CFM messages use the CFM EtherType, and carry the CFM maintenance level for the domain to which they apply.

This section describes the following CFM messages:

Continuity Check (IEEE 802.1ag and ITU-T Y.1731)

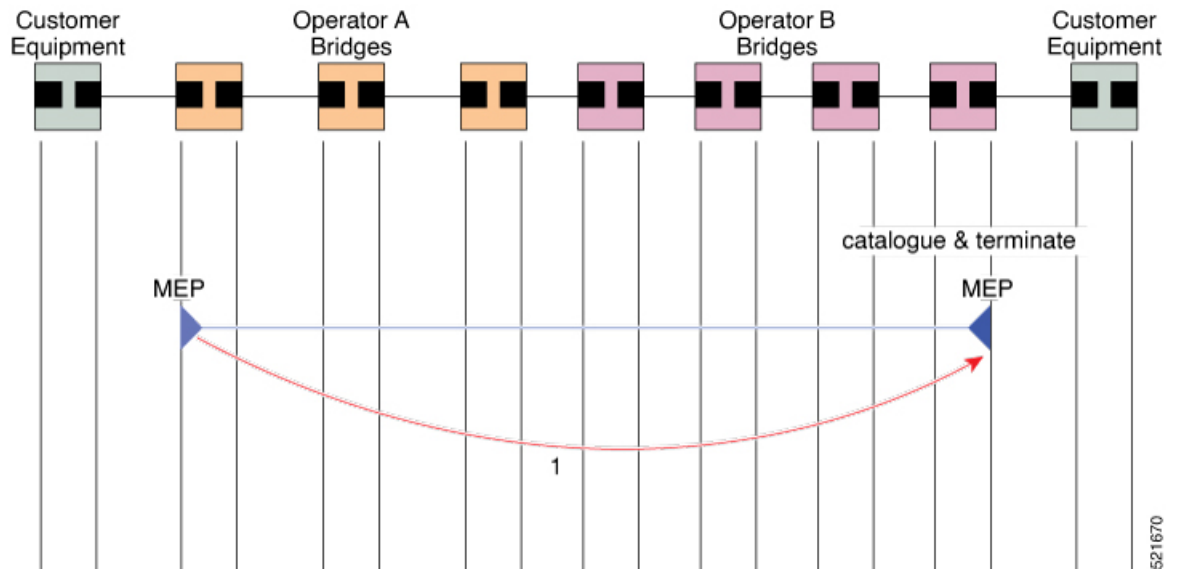
Continuity Check Messages (CCMs) are “heartbeat” messages exchanged periodically between all the MEPs in a service. Each MEP sends out multicast CCMs, and receives CCMs from all the other MEPs in the service—these are referred to as *peer MEPs*. This allows each MEP to discover its peer MEPs, and to verify that there is connectivity between them.

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
Continuity Check (IEEE 802.1ag and ITU-T Y.1731)	Release 25.3.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) This feature is now supported on: • 8011-4G24Y4H-I • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
Continuity Check (IEEE 802.1ag and ITU-T Y.1731)	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) Maintenance Endpoints (MEPs) capable of supporting Ethernet OAM send Continuity Check Messages (CCMs) at regular intervals to verify connectivity. If one of them stops receiving CCMs from another endpoint, it infers that there is a connectivity issue or fault in the path. This feature is now supported on: • 8011-4G24Y4H-I

MIPs also receive CCMs. MIPs use the information to build a MAC learning database that is used when responding to Linktrace. For more information about Linktrace, see the [“Linktrace \(IEEE 802.1ag and ITU-T Y.1731\)”](#) section.

Figure 5: Continuity Check Message Flow



All the MEPs in a service must transmit CCMs at the same interval. IEEE 802.1ag defines the following possible intervals that can be used:

- 100 ms (only supported on bundle members)
- 1 s
- 10 s
- 1 minute
- 10 minutes

A MEP detects a loss of connectivity with one of its peer MEPs when some number of CCMs are missed. This occurs when sufficient time has passed during which a certain number of CCMs were expected, given the CCM interval. This number is called the *loss threshold*, and is usually set to 3.

With the exception of bundle members, CFM is supported only on interfaces that have Layer 2 transport feature enabled.

CCM messages carry a variety of information that allows different defects to be detected in the service. This information includes:

- A configured identifier for the domain of the transmitting MEP. This is referred to as the Maintenance Domain Identifier (MDID).
- A configured identifier for the service of the transmitting MEP. This is referred to as the Short MA Name (SMAN). Together, the MDID and the SMAN make up the Maintenance Association Identifier (MAID). The MAID must be configured identically on every MEP in the service.
- These are restrictions on the type of MAID that are supported for sessions with time interval of less than 1 minute. The MAID supports two types of formats on offloaded MEPs:
 - No Domain Name Format
 - MD Name Format = 1-NoDomainName

- Short MA Name Format = 3 - 2 bytes integer value
- Short MA Name Length = 2 - fixed length
- Short MA Name = 2 bytes of integer
- 1731 Maid Format
 - MD Name Format = 1-NoDomainName
 - MA Name Format(MEGID Format) = 32
 - MEGID Length = 13 - fixed length
 - MEGID(ICCCode) = 6 Bytes
 - MEGID(UMC) = 7 Bytes
 - ITU Carrier Code (ICC) - Number of different configurable ICC code - 15 (for each NPU)
 - Unique MEG ID Code (UMC) - 4

Maintenance Association Identifier (MAID) comprises of the Maintenance Domain Identifier (MDID) and Short MA Name (SMAN). MDID only supports **null** value and SMAN only supports ITU Carrier Code (ICC) or a numerical. No other values are supported.

- An example for configuring domain ID null is: **ethernet cfm domain SMB level 3 id null**
- An example for configuring SMAN is: **ethernet cfm domain SMB level 3 id null service 901234AB xconnect group 99999 p2p 99999 id number 1**
- A configured numeric identifier for the MEP (the MEP ID). Each MEP in the service must be configured with a different MEP ID.
- Dynamic Remote MEPs are not supported for MEPs with less than 1 min interval. You must configure MEP CrossCheck for all such MEPs.
- Sequence numbering is not supported for MEPs with less than 1 minute interval.
- In a Remote Defect Indication (RDI), each MEP includes this in the CCMs it is sending, if it has detected a defect relating to the CCMs it is receiving. This notifies all the MEPs in the service that a defect has been detected somewhere in the service.
- The interval at which CCMs are being transmitted.
- CCM Tx/Rx statistics counters are not supported for MEPs with less than 1 minute intervals.
- Sender TLV and Cisco Proprietary TLVs are not supported for MEPs with less than 1 minute intervals.
- The status of the interface where the MEP is operating, for example, whether the interface is up, down, STP blocked, and so on.



Note The status of the interface (up/down) should not be confused with the direction of any MEPs on the interface (Up MEPs/Down MEPs).

These defects can be detected from the received CCMs:

- Interval mismatch: The CCM interval in the received CCM does not match the interval that the MEP is sending CCMs.
- Level mismatch: A MEP has received a CCM carrying a lower maintenance level than the MEPs own level.
- Loop: A CCM is received with the source MAC address equal to the MAC address of the interface where the MEP is operating.
- Configuration error: A CCM is received with the same MEP ID as the MEP ID configured for the receiving MEP.
- Cross-connect: A CCM is received with a MAID that does not match the locally configured MAID. This generally indicates a VLAN misconfiguration within the network, such that CCMs from one service are leaking into a different service.
- Peer interface down: A CCM is received that indicates the interface on the peer is down.
- Remote defect indication: A CCM is received carrying a remote defect indication.



Note This defect does not cause the MEP to include a remote defect indication in the CCMs that it is sending.

Out-of-sequence CCMs can also be detected by monitoring the sequence number in the received CCMs from each peer MEP. However, this is not considered a CCM defect.

Loopback (IEEE 802.1ag and ITU-T Y.1731)

Loopback Messages (LBM) and Loopback Replies (LBR) are used to verify connectivity between a local MEP and a particular remote MP. At the request of the administrator, a local MEP sends unicast LBMs to the remote MP.

Table 5: Feature History Table

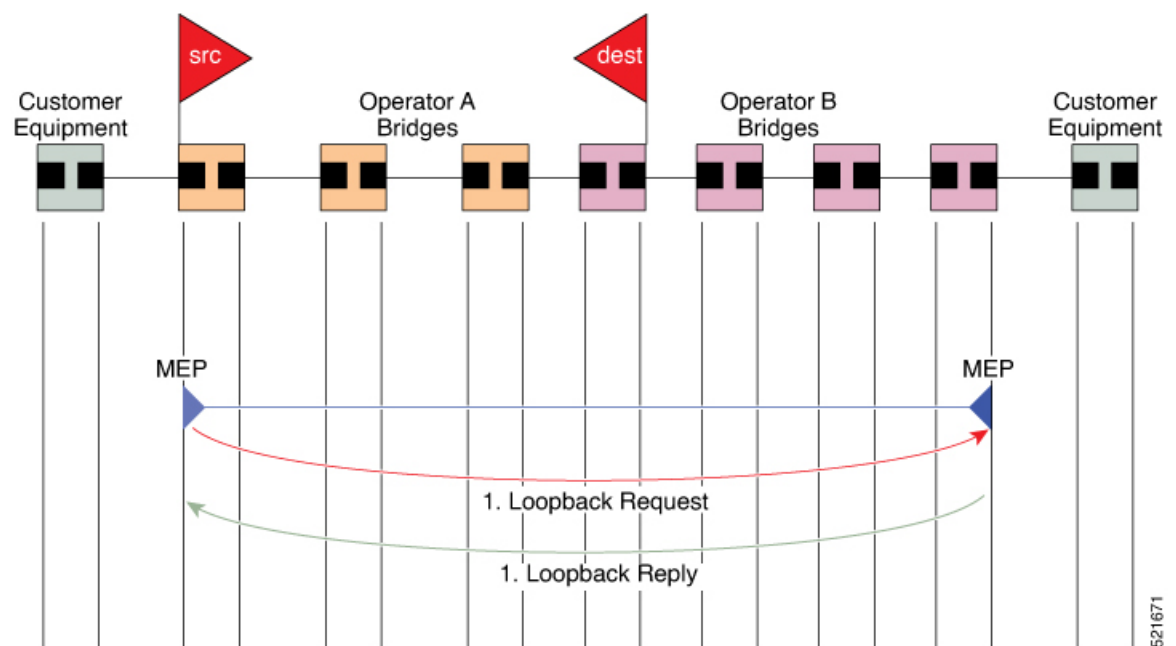
Feature Name	Release Information	Feature Description
Loopback (IEEE 802.1ag and ITU-T Y.1731)	Release 25.3.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) This feature is now supported on: • 8011-4G24Y4H-I • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E

Feature Name	Release Information	Feature Description
Loopback (IEEE 802.1ag and ITU-T Y.1731)	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) You can test connectivity between a local Maintenance End Point (MEP) and a remote Maintenance Point (MP) by sending Loopback Messages (LBM) and Loopback Replies (LBR). The connectivity verification facilitates effective network troubleshooting and maintenance without requiring a detailed hop-by-hop path analysis. This feature is now supported on: • 8011-4G24Y4H-I

On receiving each LBM, the target maintenance point sends an LBR back to the originating MEP. Loopback indicates whether the destination is reachable or not—it does not allow hop-by-hop discovery of the path. It is similar in concept to an ICMP Echo (ping). Since loopback messages are destined for unicast addresses, they are forwarded like normal data traffic, while observing the maintenance levels. At each device that the loopback reaches, if the outgoing interface is known (in the bridge's forwarding database), then the frame is sent out on that interface. If the outgoing interface is not known, then the message is flooded on all interfaces.

This figure shows an example of CFM loopback message flow between a MEP and MEP.

Figure 6: Loopback Messages



Loopback messages can be padded with user-specified data. This allows data corruption to be detected in the network. They also carry a sequence number which allows for out-of-order frames to be detected.

Linktrace (IEEE 802.1ag and ITU-T Y.1731)

Linktrace Messages (LTM) and Linktrace Replies (LTR) are used to track the path (hop-by-hop) to a unicast destination MAC address.

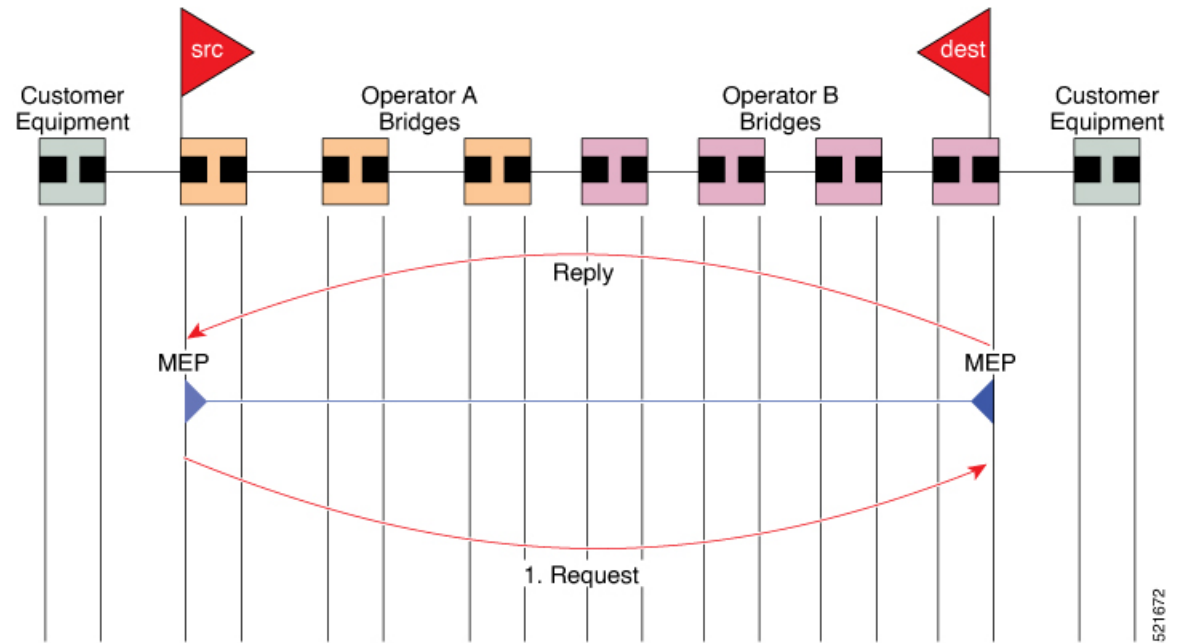
Table 6: Feature History Table

Feature Name	Release Information	Feature Description
Linktrace (IEEE 802.1ag and ITU-T Y.1731)	Release 25.3.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100], 8700 [ASIC: P100])(select variants only*); Modular Systems (8800 [LC ASIC: P100])(select variants only*) This feature is now supported on: • 8011-4G24Y4H-I • 8711-32FH-M • 88-LC1-52Y8H-EM • 88-LC1-12TH24FH-E
Linktrace (IEEE 802.1ag and ITU-T Y.1731)	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) You can trace the path to a specific unicast destination MAC address on a network using the Linktrace Messages (LTM) and Linktrace Replies (LTR) diagnostic tools. These tools perform a detailed hop-by-hop path analysis to identify connectivity problems. This feature is now supported on: • 8011-4G24Y4H-I

At the request of the operator, a local MEP sends an Linktrace Messages (LTM). Each hop where there is a maintenance point sends an Linktrace Replies (LTR) back to the originating MEP. This allows the administrator to discover connectivity data about the path. It is similar in concept to IP traceroute, although the mechanism is different. In IP traceroute, successive probes are sent, whereas CFM Linktrace uses a single LTM which is forwarded by each MP in the path. LTMs are multicast, and carry the unicast target MAC address as data within the frame. They are intercepted at each hop where there is a maintenance point, and either retransmitted or dropped to discover the unicast path to the target MAC address.

This figure shows an example of CFM linktrace message flow between MEPs and MEPs.

Figure 7: Linktrace Message Flow



The linktrace mechanism is designed to provide useful information even after a network failure. This allows it to be used to locate failures, for example after a loss of continuity is detected. To achieve this, each MP maintains a CCM Learning Database. This maps the source MAC address for each received CCM to the interface through which the CCM was received. It is similar to a typical bridge MAC learning database, except that it is based only on CCMs and it times out much more slowly—on the order of days rather than minutes.



Note In IEEE 802.1ag, the CCM Learning Database is referred to as the MIP CCM Database. However, it applies to both MIPs and MEPs.

In IEEE 802.1ag, when an MP receives an LTM message, it determines whether to send a reply using the following steps:

1. The target MAC address in the LTM is looked up in the bridge MAC learning table. If the MAC address is known, and therefore the egress interface is known, then an LTR is sent.
2. If the MAC address is not found in the bridge MAC learning table, then it is looked up in the CCM learning database. If it is found, then an LTR is sent.
3. If the MAC address is not found, then no LTR is sent (and the LTM is not forwarded).

If the target MAC has never been seen previously in the network, the linktrace operation will not produce any results.



Note IEEE 802.1ag and ITU-T Y.1731 define slightly different linktrace mechanisms. In particular, the use of the CCM learning database and the algorithm described above for responding to LTM messages are specific to IEEE 802.1ag. IEEE 802.1ag also specifies additional information that can be included in LTRs. Regardless of the differences, the two mechanisms are interoperable.

Configurable Logging

CFM supports logging of various conditions to syslog. Logging can be enabled independently for each service, and when the following conditions occur:

- New peer MEPs are detected, or loss of continuity with a peer MEP occurs.
- Changes to the CCM defect conditions are detected.
- Cross-check “missing” or “unexpected” conditions are detected.
- AIS condition detected (AIS messages received) or cleared (AIS messages no longer received).
- EFD used to shut down an interface, or bring it back up.

Unidirectional link detection protocol

Unidirectional link detection (UDLD) is a single-hop physical link protocol for monitoring an ethernet link, including both point-to-point and shared media links. This is a Cisco-proprietary Ethernet OAM protocol that detects link error conditions such as miswiring or unidirectional link failure, which are not detected at the physical link layer. This protocol is specifically useful for identifying potential wiring errors caused when using unbundled fiber links, that can lead to a mismatch between the transmitting and receiving connections of a port.

Table 7: Feature History Table

Feature Name	Release	Description
Unidirectional link detection protocol support on physical Ethernet interfaces	Release 25.2.1	Introduced in this release on: Fixed Systems (8700 [ASIC:K100]) (select variants only*) This release supports: • Unidirectional Link Detection Protocol (UDLD) on the physical ethernet interfaces. This feature helps detect faults and miswiring conditions with unbundled fiber links and enables each device to understand its own connections as well as those of its neighbors. *This feature is now supported on Cisco 8712-MOD-M routers.
Unidirectional link detection protocol support on physical Ethernet interfaces	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) (select variants only*) * This feature is supported on: • 8011-4G24Y4H-I

Feature Name	Release	Description
Unidirectional link detection protocol support on physical Ethernet interfaces	Release 24.4.1	Introduced in this release on: Fixed Systems(8200, 8700)(select variants only*); Modular Systems (8800 [LC ASIC: P100]). The Unidirectional Link Detection Protocol (UDLD) is now supported on the Physical Ethernet interfaces on the Cisco Silicon One P100 ASIC-based Systems. This feature helps detect faults and miswiring conditions with unbundled fiber links and enables each device to understand its own connections as well as those of its neighbors. *This feature is supported on: • 8212-48FH-M • 8711-32FH-M • 8712-MOD-M This feature introduces these changes: CLI: • clear ethernet udld statistics • ethernet udld reset interface • show ethernet udld interfaces • show ethernet udld statistics

UDLD operation

UDLD exchanges protocol packets between the neighboring devices. UDLD works if both devices on the link support UDLD and have it enabled on respective ports.

UDLD sends an initial PROBE message on the ports where it is configured. When it receives a PROBE message, UDLD sends periodic ECHO (hello) messages. Both the messages help identify the sender and its port, and also contain some information about the operating parameters of the protocol on that port. The messages also contain the device and port identifiers on the port for any neighbor devices that the local device has connected with on the port. Similarly, each device gets to know where it is connected and where its neighbors are connected. This helps in detecting faults and miswiring conditions.

The protocol employs a mechanism where information from neighbors that is not periodically refreshed is eventually timed out for fault detection.

The protocol uses a FLUSH message to indicate when UDLD is disabled on a port. This causes the peers to remove the local device from their neighbor cache to prevent a time out.

If a problem is detected, UDLD disables the affected interface and notifies the user to avoid further network problems beyond traffic loss. Example: Loops which are not detected or prevented by Spanning Tree Protocol (STP).

Types of fault detection

UDLD can detect these types of faults:

- **Transmit faults** — These are transmission failures from the local port to the peer device which also includes the faults caused by physical link failure or packet path issues on the local or peer device. These failures can lead to serious network issues such as loops which occur specifically when a link is unidirectional.
- **Miswiring faults** — These are instances that occur when using unbundled fibers to connect fiber optic ports. In such instances, the receiving and transmitting sides of a port on the local device are connected to different peer ports (on the same device or on different devices).
- **Loopback faults** — In these instances, the receiving and transmitting sides of a port are connected to each other, creating a loopback condition. This can be an intentional mode of operation, for certain types of testing, but UDLD must not be used in these cases.
- **Receive faults** — The protocol uses a heartbeat signal that is transmitted at a negotiated periodic interval to the peer device. Missed heartbeats can therefore be used to detect failures on the receiving side of the link (where they do not result in interface state changes). These could be caused by a unidirectional link with a failure only affecting the receiving side, or by a link which has developed a bidirectional fault. This detection depends on reliable, regular packet transmission by the peer device. For this reason, the UDLD protocol has two configurable modes of operation namely **Normal** mode and **Aggressive** mode, which determine the behavior on a heartbeat timeout. For more information about these modes, see [UDLD modes of operation, on page 27](#).

UDLD modes of operation

UDLD can operate in these modes:

- **Normal mode:** In this mode, if a `Receive Fault` is detected, the user is informed and no further action is taken.
- **Aggressive mode:** In this mode, if a `Receive Fault` is detected, the user is informed and the affected port is disabled.

UDLD aging mechanism

Aging of UDLD information occurs in a `Receive Fault` condition when the port that runs UDLD does not receive UDLD packets from the neighbor port for a duration of the hold time. The hold time for the port is dictated by the remote port and is dependent on the message interval at the remote end. The shorter the message interval, the shorter is the hold time and faster the detection of the fault. The hold time is three times the message interval in Cisco IOS XR Software.

UDLD information can age out due to the high error rate on the port caused by a physical issue or duplex mismatch. Packet drops due to age out does not mean that the link is unidirectional. UDLD in normal mode does not disable such link.

It is important to choose the right message interval to ensure proper detection time. The message interval should be fast enough to detect the unidirectional link before the forwarding loop is created. The default message interval is 60 seconds. The detection time is approximately equal to three times the message interval. Therefore, when using default UDLD timers, UDLD does not timeout the link faster than the STP aging time.

UDLD state machines

UDLD uses two types of finite state machines (FSMs), generally referred as state machines. The Main FSM deals with all the phases of operation of the protocol while the Detection FSM handles only the phases that determine the status of a port.

Main FSM

The Main FSM can be in one of these states:

- **Init:** Protocol is initializing.
- **UDLD inactive:** Port is down or UDLD is disabled.
- **Linkup:** Port is up and running, and UDLD is in the process of detecting a neighbor.
- **Detection:** A hello message from a new neighbor has been received and the Detection FSM is running to determine the status of the port.
- **Advertisement:** The Detection FSM has run and concluded that the port is operating correctly, periodic hello messages are being sent and the hello messages from neighbors are monitored.
- **Port shutdown:** The Detection FSM detected a fault, or all neighbors were timed out in Aggressive mode, and the port has been disabled as a result.

Detection FSM

The Detection FSM can be in one of these states:

- **Unknown:** Detection has not yet been performed or UDLD has been disabled.
- **Unidirectional detected:** A unidirectional link condition has been detected because a neighbor does not see the local device, the port will be disabled.
- **Tx/Rx loop:** A loopback condition has been detected by receiving a type, length, and value (TLV) message with the ports own identifiers, the port will be disabled.
- **Neighbor mismatch:** A miswiring condition has been detected in which a neighbor can identify other devices than the devices the local device can see and the port will be disabled.
- **Bidirectional detected:** UDLD hello messages are exchanged successfully in both the directions, the port is operating correctly.

Limitations

- UDLD on Cisco 8000 Series Routers does not work if the peer UDLD configuration has custom MAC address; Peer must have either Cisco MAC address or IEEE Slow Proto MAC address.
- Use only these MAC Addresses to establish a successful connection and communication with the Cisco 8000 Series Routers.
 - cisco-l2cp (0x01000ccccccc) - Cisco proprietary MAC Address which can also be used by all other Cisco protocols.
 - ieee-slow-protocols (0x0180c2000002) - IEEE Slow Protocol MAC Address.
- UDLD is not tunneled through L2VPN like other slow protocols.
- UDLD must not be enabled on a Switched Port Analyzer (SPAN) source or a destination port.
- The UDLD protocol is not supported on the subinterfaces and bundle interfaces.

Configure UDLD

SUMMARY STEPS

1. **configure**
2. **interface** [**GigabitEthernet** | **TenGigE**] *interface-path-id*
3. **ethernet udld**
4. **mode** {**normal** | **aggressive**}
5. **message-time**
6. **logging disable**
7. **end**

DETAILED STEPS

Procedure

Step 1

configure

Example:

```
RP/0/RSP0/CPU0:router# configure
```

Enters global configuration mode.

Step 2

interface [**GigabitEthernet** | **TenGigE**] *interface-path-id*

Example:

```
RP/0/RSP0/CPU0:router(config)# interface
TenGigE 0/1/0/0
```

Enters interface configuration mode and specifies the Ethernet interface name and notation *rack/slot/module/port*.

Note

The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.

Step 3 **ethernet udld****Example:**

```
RP/0/RSP0/CPU0:router(config-if)# ethernet udld
```

Enables ethernet UDLD functionality and enters interface Ethernet UDLD configuration mode.

Step 4 **mode {normal |aggressive}****Example:**

```
RP/0/RSP0/CPU0:router(config-if-udld)# mode normal
```

(Optional) Specifies the mode of operation for UDLD. The options are normal and aggressive.

Step 5 **message-time****Example:**

```
RP/0/RSP0/CPU0:router(config-if-udld)# message-time 70
```

(Optional) Specifies the message time (in seconds) to use for the UDLD protocol. The value ranges from 7 to 90 seconds.

Step 6 **logging disable****Example:**

```
RP/0/RSP0/CPU0:router(config-if-udld)# logging disable
```

(Optional) Suppresses the operational UDLD syslog messages.

Step 7 **end****Example:**

```
RP/0/RSP0/CPU0:router(config-if-udld)# end
```

Ends the configuration session and exits to the EXEC mode.

How to Configure Ethernet OAM

This section provides these configuration procedures:

Configuring Ethernet OAM

Custom EOAM settings can be configured and shared on multiple interfaces by creating an EOAM profile in Ethernet configuration mode and then attaching the profile to individual interfaces. The profile configuration does not take effect until the profile is attached to an interface. After an EOAM profile is attached to an interface, individual EOAM features can be configured separately on the interface to override the profile settings when desired.

This section describes how to configure an EOAM profile and attach it to an interface in these procedures:

Configuring an Ethernet OAM Profile

Perform these steps to configure an Ethernet OAM profile.

Procedure

	Command or Action	Purpose
Step 1	configure Example: <pre>RP/0/RP0/CPU0:router# configure terminal</pre>	Enters global configuration mode.
Step 2	ethernet oam profile <i>profile-name</i> Example: <pre>RP/0/RP0/CPU0:router(config)# ethernet oam profile Profile_1</pre>	Creates a new Ethernet Operations, Administration and Maintenance (OAM) profile and enters Ethernet OAM configuration mode.
Step 3	link-monitor Example: <pre>RP/0/RP0/CPU0:router(config-eoam)# link-monitor</pre>	Enters the Ethernet OAM link monitor configuration mode.
Step 4	symbol-period window { <i>milliseconds window</i> <i>symbols window</i> [<i>thousand</i> <i>million</i> <i>billion</i>] } Example: <pre>RP/0/RP0/CPU0:router(config-eoam-lm)# symbol-period window 60000</pre>	(Optional) Configures the window size for an Ethernet OAM symbol-period error event. If specified in milliseconds, the range is 1000 to 60000. If not specified as a multiple of 1 second, the actual window used will be rounded up to the nearest second, with thresholds scaled accordingly. If specified in symbols, the range is interface speed dependent (must be between the maximum number of symbols that could be received in 1 second and the maximum number of symbols that could be received in 1 minute). Again the actual window used is rounded up to the nearest second, with thresholds scaled accordingly. The default value is 1000 milliseconds.
Step 5	symbol-period threshold { <i>ppm</i> [<i>low threshold</i>] [<i>high threshold</i>] <i>symbols</i> [<i>low threshold</i> [<i>thousand</i> <i>million</i> <i>billion</i>]] [<i>high threshold</i> [<i>thousand</i> <i>million</i> <i>billion</i>]] } Example: <pre>RP/0/RP0/CPU0:router(config-eoam-lm)# symbol-period threshold ppm low 100 high 1000000</pre>	(Optional) Configures the thresholds that trigger an Ethernet OAM symbol-period error event, in symbols or ppm (errors per million symbols). When using this command at least one of the high and low thresholds must be specified. If the low threshold is not specified, the default value is used. If the high threshold is not specified, no action is performed in response to an event. The high threshold must not be smaller than the low threshold.

	Command or Action	Purpose
		If specified in ppm, the range (for both thresholds) is 1 to 1000000. If specified in symbols, the range (for both thresholds) is 1 to the maximum window size in symbols, see symbol-period window. The default low threshold is 1 symbol.
Step 6	frame window milliseconds window Example: <pre>RP/0/RP0/CPU0:router(config-eoam-lm)# frame window milliseconds 60</pre>	(Optional) Configures the frame window size (in milliseconds) of an OAM frame error event. The range is from 1000 to 60000. The default value is 1000.
Step 7	frame threshold [low threshold] [high threshold] Example: <pre>RP/0/RP0/CPU0:router(config-eoam-lm)# frame threshold low 10000000 high 60000000</pre>	(Optional) Configures the thresholds (in symbols) that triggers an Ethernet OAM frame error event. When using this command at least one of the high and low thresholds must be specified. If the low threshold is not specified, the default value is used. If the high threshold is not specified, no action is performed in response to an event. The high threshold must not be smaller than the low threshold. The range is from 1 to 60000000. The default low threshold is 1.
Step 8	frame-period window { milliseconds window frames window [thousand million billion] } Example: <pre>RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period window milliseconds 60000</pre>	(Optional) Configures the window size for an Ethernet OAM frame-period error event. The range is from 100 to 60000, if defined in milliseconds. If the window is defined as say, 200ms, and the interface could receive at most say 10000 minimum size frames in 200ms, then the actual window size used will be the time taken to receive 10000 frames, rounded up to the nearest second. The thresholds will be scaled accordingly. If specified in frames, the range is interface speed dependent, but must be between the number of minimum size frames that could be received in 100ms and the number of minimum size frames that could be received in 1 minute. If the window is defined as 20000 frames, the actual window size used will be the time taken to receive 20000 frames, rounded up to the nearest second. The thresholds will be scaled accordingly. The default value is 1000 milliseconds.
Step 9	frame-period threshold { ppm [low threshold] [high threshold] frames [low threshold [thousand million billion]] [high threshold [thousand million billion]] } Example:	(Optional) Configures the thresholds (either in frames or in ppm - errors per million frames) that trigger an Ethernet OAM frame-period error event. When using this command at least one of the high and low thresholds must be specified. If the low threshold is not specified, the default value is used. If the high threshold is not specified, no

	Command or Action	Purpose
	<pre>RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period threshold ppm low 100 high 1000000</pre>	action is performed in response to an event. The high threshold must not be smaller than the low threshold. The range for both thresholds is from 1 to 1000000 if specified in ppm. If specified in frames, the range is from 1 to the maximum frame-period window size in frames, see frame-period window. The default low threshold is 1 ppm.
Step 10	frame-seconds window milliseconds window Example: <pre>RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds window milliseconds 900000</pre>	(Optional) Configures the window size (in milliseconds) for the OAM frame-seconds error event. The range is 10000 to 900000. The default value is 6000.
Step 11	frame-seconds threshold [low threshold] [high threshold] Example: <pre>RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds threshold low 3 threshold high 900</pre>	(Optional) Configures the thresholds (in seconds) that trigger a frame-seconds error event. When using this command at least one of the high and low thresholds must be specified. If the low threshold is not specified, the default value is used. If the high threshold is not specified, no action is performed in response to an event. The high threshold must not be smaller than the low threshold. The range is 1 to 900 The default value is 1.
Step 12	exit Example: <pre>RP/0/RP0/CPU0:router(config-eoam-lm)# exit</pre>	Exits back to Ethernet OAM mode.
Step 13	mib-retrieval Example: <pre>RP/0/RP0/CPU0:router(config-eoam)# mib-retrieval</pre>	Enables MIB retrieval in an Ethernet OAM profile or on an Ethernet OAM interface.
Step 14	connection timeout <timeout> Example: <pre>RP/0/RP0/CPU0:router(config-eoam)# connection timeout 30</pre>	Configures the connection timeout period for an Ethernet OAM session, as a multiple of the hello interval. The range is 2 to 30. The default value is 5.
Step 15	hello-interval 1s Example: <pre>RP/0/RP0/CPU0:router(config-eoam)# hello-interval 1s</pre>	Configures the time interval between hello packets for an Ethernet OAM session. The default is 1 second (1s).

	Command or Action	Purpose
Step 16	mode {active passive} Example: RP/0/RP0/CPU0:router(config-eoam)# mode passive	Configures the Ethernet OAM mode. The default is active.
Step 17	require-remote mode {active passive} Example: RP/0/RP0/CPU0:router(config-eoam)# require-remote mode active	Requires that active mode or passive mode is configured on the remote end before the OAM session becomes active.
Step 18	require-remote mib-retrieval Example: RP/0/RP0/CPU0:router(config-eoam)# require-remote mib-retrieval	Requires that MIB-retrieval is configured on the remote end before the OAM session becomes active.
Step 19	action capabilities-conflict {disable efd error-disable-interface log} Example: RP/0/RP0/CPU0:router(config-eoam)# action capabilities-conflict efd	Specifies the action that is taken on an interface when a capabilities-conflict event occurs. The default action is to create a syslog entry.
Step 20	action critical-event {disable error-disable-interface log} Example: RP/0/RP0/CPU0:router(config-eoam)# action critical-event error-disable-interface	Specifies the action that is taken on an interface when a critical-event notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry.
Step 21	action discovery-timeout {disable efd error-disable-interface log} Example: RP/0/RP0/CPU0:router(config-eoam)# action discovery-timeout efd	Specifies the action that is taken on an interface when a connection timeout occurs. The default action is to create a syslog entry.
Step 22	action dying-gasp {disable error-disable-interface log} Example: RP/0/RP0/CPU0:router(config-eoam)# action dying-gasp error-disable-interface	Specifies the action that is taken on an interface when a dying-gasp notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry.
Step 23	action high-threshold {disable error-disable-interface log} Example:	Specifies the action that is taken on an interface when a high threshold is exceeded. The default is to take no action when a high threshold is exceeded.

Configuring Ethernet OAM

Attaching an Ethernet OAM Profile to an Interface

Perform these steps to attach an Ethernet OAM profile to an interface:

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure terminal	Enters global configuration mode.
Step 2	interface [FastEthernet HundredGigE TenGigE] <i>interface-path-id</i> Example: RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0	Enters interface configuration mode and specifies the Ethernet interface name and notation <i>rack/slot/module/port</i> . Note The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.
Step 3	ethernet oam Example: RP/0/RP0/CPU0:router(config-if)# ethernet oam	Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.
Step 4	profile profile-name Example: RP/0/RP0/CPU0:router(config-if-eoam)# profile Profile_1	Attaches the specified Ethernet OAM profile (<i>profile-name</i>), and all of its configuration, to the interface.
Step 5	commit Example: RP/0/RP0/CPU0:router(config-if)# commit	Saves the configuration changes to the running configuration file and remains within the configuration session.
Step 6	end Example: RP/0/RP0/CPU0:router(config-if)# end	Ends the configuration session and exits to the EXEC mode.

Configuring Ethernet OAM at an Interface and Overriding the Profile Configuration

Using an EOAM profile is an efficient way of configuring multiple interfaces with a common EOAM configuration. However, if you want to use a profile but also change the behavior of certain functions for a particular interface, then you can override the profile configuration. To override certain profile settings that are applied to an interface, you can configure that command in interface Ethernet OAM configuration mode to change the behavior for that interface.

In some cases, only certain keyword options are available in interface Ethernet OAM configuration due to the default settings for the command. For example, without any configuration of the **action** commands, several forms of the command have a default behavior of creating a syslog entry when a profile is created and applied to an interface. Therefore, the **log** keyword is not available in Ethernet OAM configuration for these commands in the profile because it is the default behavior. However, the **log** keyword is available in Interface Ethernet OAM configuration if the default is changed in the profile configuration so you can retain the action of creating a syslog entry for a particular interface.

To see all of the default Ethernet OAM configuration settings, see the [“Verifying the Ethernet OAM Configuration” section](#).

To configure Ethernet OAM settings at an interface and override the profile configuration, perform these steps:

SUMMARY STEPS

1. **configure**
2. **interface** [HundredGigE | TenGigE] *interface-path-id*
3. **ethernet oam**
4. *interface-Ethernet-OAM-command*
5. **commit**
6. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure Example: <pre>RP/0/RP0/CPU0:router# configure terminal</pre>	Enters global configuration mode.
Step 2	interface [HundredGigE TenGigE] <i>interface-path-id</i> Example: <pre>RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/0</pre>	Enters interface configuration mode and specifies the Ethernet interface name and notation <i>rack/slot/module/port</i> . Note The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.
Step 3	ethernet oam Example: <pre>RP/0/RP0/CPU0:router(config-if)# ethernet oam</pre>	Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.
Step 4	<i>interface-Ethernet-OAM-command</i> Example: <pre>RP/0/RP0/CPU0:router(config-if-eoam)# action capabilities-conflict error-disable-interface</pre>	Configures a setting for an Ethernet OAM configuration command and overrides the setting for the profile configuration, where <i>interface-Ethernet-OAM-command</i> is one of the supported commands on the platform in interface Ethernet OAM configuration mode.

	Command or Action	Purpose
Step 5	commit Example: RP/0/RP0/CPU0:router(config-if)# commit	Saves the configuration changes to the running configuration file and remains within the configuration session.
Step 6	end Example: RP/0/RP0/CPU0:router(config-if)# end	Ends the configuration session and exits to the EXEC mode.

Verifying the Ethernet OAM Configuration

Use the **show ethernet oam configuration** command to display the values for the Ethernet OAM configuration for a particular interface, or for all interfaces. The following example shows the default values for Ethernet OAM settings:

```
RP/0/RP0/CPU0:router# show ethernet oam configuration
Thu Aug  5 22:07:06.870 DST
GigabitEthernet0/4/0/0:
  Hello interval:                               1s
  Mib retrieval enabled:                         N
  Uni-directional link-fault detection enabled:  N
  Configured mode:                             Active
  Connection timeout:                           5
  Symbol period window:                         0
  Symbol period low threshold:                   1
  Symbol period high threshold:                 None
  Frame window:                                 1000
  Frame low threshold:                          1
  Frame high threshold:                         None
  Frame period window:                          1000
  Frame period low threshold:                    1
  Frame period high threshold:                  None
  Frame seconds window:                         60000
  Frame seconds low threshold:                   1
  Frame seconds high threshold:                 None
  High threshold action:                        None
  Link fault action:                            Log
  Dying gasp action:                            Log
  Critical event action:                        Log
  Discovery timeout action:                     Log
  Capabilities conflict action:                 Log
  Wiring conflict action:                       Error-Disable
  Session up action:                            Log
  Session down action:                         Log
  Require remote mode:                          Ignore
  Require remote MIB retrieval:                 N
```

Configuring Ethernet CFM

To configure Ethernet CFM, perform the following tasks:



Note CFM is not supported for the following:

- L3 Interfaces and Sub-Interfaces
- Bridge Domain, Release 7.3.1 and earlier
- VPLS, Release 7.3.1 and earlier

Configuring a CFM Maintenance Domain

To configure a CFM maintenance domain, perform the following steps:

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** **null**] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
4. **traceroute cache hold-time** *minutes* **size** *entries*
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	ethernet cfm Example: RP/0/RP0/CPU0:router(config)# ethernet cfm	Enters Ethernet Connectivity Fault Management (CFM) configuration mode.
Step 3	domain <i>domain-name</i> level <i>level-value</i> [id null] [dns <i>DNS-name</i>] [mac <i>H.H.H</i>] [string <i>string</i>]] Example: RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1	Creates and names a container for all domain configurations and enters CFM domain configuration mode. The level must be specified. The id is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.

	Command or Action	Purpose
Step 4	traceroute cache hold-time <i>minutes</i> size <i>entries</i> Example: <pre>RP/0/RP0/CPU0:router(config-cfm)# traceroute cache hold-time 1 size 3000</pre>	(Optional) Sets the maximum limit of traceroute cache entries or the maximum time limit to hold the traceroute cache entries. The default is 100 minutes and 100 entries.
Step 5	end or commit Example: <pre>RP/0/RP0/CPU0:router(config-cfm-dmn)# commit</pre>	Saves configuration changes. - When you use the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</pre> - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. - Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring services for a CFM maintenance domain

From Release 24.4.1, Cisco 8000 routers support 500 CFM sessions.

To configure services for a CFM maintenance domain, perform the following steps:

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** *[null]*] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
4. **service** *service-name* {**down-meps** | **xconnect group** *xconnect-group-name* **p2p** *xconnect-name*} [**id** *[icc-based icc-string umc-string]*] | [**number** *number*]
5. **end or commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	ethernet cfm Example: RP/0/RP0/CPU0:router(config)# ethernet cfm	Enters Ethernet CFM configuration mode.
Step 3	domain domain-name level level-value [id [null]] [dns DNS-name] [mac H.H.H] [string string]] Example: RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1	Creates and names a container for all domain configurations at a specified maintenance level, and enters CFM domain configuration mode. The id is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	service service-name {down-meps xconnect group xconnect-group-name p2p xconnect-name} [id [icc-based icc-string umc-string]] [[number number] Example: RP/0/RP0/CPU0:router(config-cfm-dmn)# service xconnect group X1	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs. The id sets the short MA name.
Step 5	end or commit Example: RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit	Saves configuration changes. - When you use the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.

	Command or Action	Purpose
		Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Enabling and Configuring Continuity Check for a CFM Service

To configure Continuity Check for a CFM service, complete the following steps:

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** [null]] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
4. **service** *service-name* {**down-meps** | **xconnect group** *xconnect-group-name* **p2p** *xconnect-name*} [**id** [**icc-based** *icc-string umc-string*] | [**number** *number*]
5. **continuity-check interval** *time* [**loss-threshold** *threshold*]
6. **continuity-check archive hold-time** *minutes*
7. **continuity-check loss auto-traceroute**
8. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure Example: <pre>RP/0/RP0/CPU0:router# configure</pre>	Enters global configuration mode.
Step 2	ethernet cfm Example: <pre>RP/0/RP0/CPU0:router(config)# ethernet cfm</pre>	Enters Ethernet Connectivity Fault Management (CFM) configuration mode.
Step 3	domain <i>domain-name</i> level <i>level-value</i> [id [null]] [dns <i>DNS-name</i>] [mac <i>H.H.H</i>] [string <i>string</i>]] Example: <pre>RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</pre>	Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The id is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.

	Command or Action	Purpose
Step 4	service service-name {down-meps xconnect group xconnect-group-name p2p xconnect-name} [id [icc-based icc-string umc-string]] [[number number] Example: <pre>RP/0/RP0/CPU0:router(config-cfm-dmn)# service xconnect group X1</pre>	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs, or associate the service with a xconnect where up MEPs will be created. The id sets the short MA name.
Step 5	continuity-check interval time [loss-threshold threshold] Example: <pre>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check interval 100m loss-threshold 10</pre>	(Optional) Enables Continuity Check and specifies the time interval at which CCMs are transmitted or to set the threshold limit for when a MEP is declared down.
Step 6	continuity-check archive hold-time minutes Example: <pre>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check archive hold-time 100</pre>	(Optional) Configures how long information about peer MEPs is stored after they have timed out.
Step 7	continuity-check loss auto-traceroute Example: <pre>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check loss auto-traceroute</pre>	(Optional) Configures automatic triggering of a traceroute when a MEP is declared down.
Step 8	end or commit Example: <pre>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit</pre>	Saves configuration changes. - When you use the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. - Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Cross-Check on a MEP for a CFM Service

To configure cross-check on a MEP for a CFM service and specify the expected set of MEPs, complete the following steps:

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** *[null]*] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
4. **service** *service-name* {**down-meps** | **xconnect group** *xconnect-group-name* **p2p** *xconnect-name*} [**id** *[icc-based* *icc-string* *umc-string*] | [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]]
5. **mep crosscheck**
6. **mep-id** *mep-id-number*
7. **end** or **commit**

DETAILED STEPS

Procedure		
	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	ethernet cfm Example: RP/0/RP0/CPU0:router# ethernet cfm	Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.
Step 3	domain <i>domain-name</i> level <i>level-value</i> [id <i>[null]</i>] [dns <i>DNS-name</i>] [mac <i>H.H.H</i>] [string <i>string</i>]] Example: RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1	Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The id is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	service <i>service-name</i> { down-meps xconnect group <i>xconnect-group-name</i> p2p <i>xconnect-name</i> } [id <i>[icc-based</i> <i>icc-string</i> <i>umc-string</i>] [string <i>text</i>] [number <i>number</i>] [vlan-id <i>id-number</i>] [vpn-id <i>oui-vpnid</i>]]	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs, or associate the service with a xconnect where up MEPs will be created. The id sets the short MA name.

	Command or Action	Purpose
Step 5	mep crosscheck Example: <pre>RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mep crosscheck mep-id 10</pre>	Enters CFM MEP crosscheck configuration mode.
Step 6	mep-id mep-id-number Example: <pre>RP/0/RP0/CPU0:router(config-cfm-xcheck)# mep-id 10</pre>	Enables cross-check on a MEP. Note - For non-offloaded and software-offloaded MEPs, use the mep-id mep-id-number [mac-address mac-address] command. - For hardware-offloaded MEPs, use the mep-id mep-id-number command. From Release 24.2.11, mac-address mac-address option is obsolete for hardware-offloaded MEPs. - Repeat this command for every MEP that you want included in the expected set of MEPs for cross-check.
Step 7	end or commit Example: <pre>RP/0/RP0/CPU0:router(config-cfm-xcheck)# commit</pre>	Saves configuration changes. - When you use the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. - Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Other Options for a CFM Service

To configure other options for a CFM service, complete the following steps:

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain** *domain-name* **level** *level-value* [**id** [null] [**dns** *DNS-name*] [**mac** *H.H.H*] [**string** *string*]]
4. **service** *service-name* {**down-meps** | **xconnect group** *xconnect-group-name* **p2p** *xconnect-name*} [**id** [**icc-based** *icc-string umc-string*] | [**string** *text*] | [**number** *number*] | [**vlan-id** *id-number*] | [**vpn-id** *oui-vpnid*]]
5. **maximum-meps** *number*
6. **log** {**ais**|**continuity-check errors**|**continuity-check mep changes**|**crosscheck errors**|**efd**}
7. **end** or **commit**

DETAILED STEPS

Procedure		
	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	ethernet cfm Example: RP/0/RP0/CPU0:router# ethernet cfm	Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.
Step 3	domain <i>domain-name</i> level <i>level-value</i> [id [null] [dns <i>DNS-name</i>] [mac <i>H.H.H</i>] [string <i>string</i>]] Example: RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1	Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The id is the maintenance domain identifier (MDID) and is used as the first part of the maintenance association identifier (MAID) in CFM frames. If the MDID is not specified, the domain name is used as the MDID by default.
Step 4	service <i>service-name</i> { down-meps xconnect group <i>xconnect-group-name</i> p2p <i>xconnect-name</i> } [id [icc-based <i>icc-string umc-string</i>] [string <i>text</i>] [number <i>number</i>] [vlan-id <i>id-number</i>] [vpn-id <i>oui-vpnid</i>]]	Configures and associates a service with the domain and enters CFM domain service configuration mode. You can specify that the service is used only for down MEPs, or associate the service with an xconnect where up MEPs will be created. The id sets the short MA name.
Step 5	maximum-meps <i>number</i> Example: RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# maximum-meps 1000	(Optional) Configures the maximum number (2 to 8190) of MEPs across the network, which limits the number of peer MEPs recorded in the database.

	Command or Action	Purpose
Step 6	log {ais continuity-check errors continuity-check mep changes crosscheck errors efd} Example: RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log continuity-check errors	(Optional) Enables logging of certain types of events.
Step 7	end or commit Example: RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit	Saves configuration changes. - When you use the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. - Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring CFM MEPs

- For every subinterface configured under a Layer 3 parent interface, you must associate a unique 802.1Q or 802.1ad tag. Else, it leads to unknown network behavior.

SUMMARY STEPS

- configure**
- interface {HundredGigE | TenGigE} interface-path-id**
- interface {HundredGigE | TenGigE | Bundle-Ether} interface-path-id12transport**
- ethernet cfm**
- mep domain domain-name service service-name mep-id id-number**
- cos cos**
- end or commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	interface {HundredGigE TenGigE} interface-path-id Example: RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1	Type of Ethernet interface on which you want to create a MEP. Enter HundredGigE or TenGigE and the physical interface or virtual interface. Note - Use the show interfaces command to see a list of all interfaces currently configured on the router. - L3 interfaces are only supported for bundle member interfaces. Else, you must enable l2transport.
Step 3	interface {HundredGigE TenGigE Bundle-Ether} interface-path-id l2transport Example: RP/0/RP0/CPU0:router(config)# interface TenGigE 0/0/0/1	Type of Ethernet interface on which you want to create a MEP. Enter HundredGigE , TenGigE , or Bundle-Ether and the physical interface or virtual interface followed by the l2transport. L2transport configures the interface as an L2 interface. Naming convention is <i>interface-path-id.subinterface</i> . The period in front of the subinterface value is required as part of the notation.
Step 4	ethernet cfm Example: RP/0/RP0/CPU0:router(config-if)# ethernet cfm	Enters interface Ethernet CFM configuration mode.
Step 5	mep domain domain-name service service-name mep-id id-number Example: RP/0/RP0/CPU0:router(config-if-cfm)# mep domain Dm1 service Sv1 mep-id 1	Creates a maintenance end point (MEP) on an interface and enters interface CFM MEP configuration mode.
Step 6	cos cos Example: RP/0/RP0/CPU0:router(config-if-cfm-mep)# cos 7	(Optional) Configures the class of service (CoS) (from 0 to 7) for all CFM packets generated by the MEP on an interface. If not configured, the CoS is inherited from the Ethernet interface. Note For Ethernet interfaces, the CoS is carried as a field in the VLAN tag. Therefore, CoS only applies to interfaces where

	Command or Action	Purpose
		packets are sent with VLAN tags. If the cos (CFM) command is executed for a MEP on an interface that does not have a VLAN encapsulation configured, it will be ignored.
Step 7	end or commit Example: <pre>RP/0/RP0/CPU0:router(config-if-cfm-mep)# commit</pre>	Saves configuration changes. - When you use the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</pre> - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. - Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring Y.1731 AIS

This section has the following step procedures:

Configuring AIS in a CFM Domain Service

Use the following procedure to configure Alarm Indication Signal (AIS) transmission for a CFM domain service and configure AIS logging.

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain name level level**
4. **service name xconnect group xconnect-group-name p2p xconnect-name**
5. **ais transmission [interval {1s|1m}][cos cos]**
6. **log ais**
7. **end or commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	ethernet cfm Example: RP/0/RP0/CPU0:router(config)# ethernet cfm	Enters Ethernet CFM global configuration mode.
Step 3	domain name level level Example: RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1	Specifies the domain and domain level.
Step 4	service name xconnect group xconnect-group-name p2p xconnect-name Example: RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 xconnect group XG1 p2p X2	Specifies the service and cross-connect group and name.
Step 5	ais transmission [interval {1s 1m}][cos cos] Example: RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7	Configures Alarm Indication Signal (AIS) transmission for a Connectivity Fault Management (CFM) domain service.
Step 6	log ais Example: RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log ais	Configures AIS logging for a Connectivity Fault Management (CFM) domain service to indicate when AIS or LCK packets are received.
Step 7	end or commit Example: RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit	Saves configuration changes. - When you issue the end command, the system prompts you to commit changes: <pre>Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]:</pre> - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.

	Command or Action	Purpose
		- Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. - Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Configuring AIS on a CFM Interface

To configure AIS on a CFM interface, perform the following steps:

SUMMARY STEPS

1. **configure**
2. **interface gigabitethernet** *interface-path-id*
3. **ethernet cfm**
4. **ais transmission up interval 1m cos** *cos*
5. **end** or **commit**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	configure Example: RP/0/RP0/CPU0:router# configure	Enters global configuration mode.
Step 2	interface gigabitethernet <i>interface-path-id</i> Example: RP/0/RP0/CPU0:router# interface TenGigE 0/0/0/2	Enters interface configuration mode.
Step 3	ethernet cfm Example: RP/0/RP0/CPU0:router(config)# ethernet cfm	Enters Ethernet CFM interface configuration mode.
Step 4	ais transmission up interval 1m cos <i>cos</i> Example:	Configures Alarm Indication Signal (AIS) transmission on a Connectivity Fault Management (CFM) interface.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config-if-cfm)# ais transmission up interval 1m cos 7	
Step 5	end or commit Example: RP/0/RP0/CPU0:router(config-sla-prof-stat-cfg)# commit	Saves configuration changes. - When you issue the end command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting (yes/no/cancel)? [cancel]: - Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode. - Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes. - Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes. - Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.

Verifying the CFM Configuration

To verify the CFM configuration, use one or more of the following commands:

show ethernet cfm configuration-errors [domain domain-name] [interface interface-path-id]	Displays information about errors that are preventing configured CFM operations from becoming active, as well as any warnings that have occurred.
show ethernet cfm local maintenance-points domain name [service name] interface type interface-path-id] [mep mip]	Displays a list of local maintenance points.



Note After you configure CFM, the error message, *cfmd[317]: %L2-CFM-5-CCM_ERROR_CCMS_MISSED : Some received CCMs have not been counted by the CCM error counters*, may display. This error message does not have any functional impact and does not require any action from you.

CFM Over Bundles

CFM over bundle supports the following:

- CFM Maintenance Points — UP MEP, Down MEP, which only includes L2 bundle main and sub-interfaces.
- CCM interval of 100 ms, 1s, 10s, 1min, and 10mins.
- RP OIR/VM reload without impacting learnt CFM peer MEPs.
- Process restart without impacting CFM sessions.
- Static MEPs.

Restrictions for Configuration of CFM on Bundles

Following are the restrictions for configuring CFM over bundle member interfaces:

- Only Layer 2 bundle Ethernet interfaces and sub-interfaces are supported, which are part of a L2VPN cross-connect.
- No support for 3.3ms and 10ms CCM interval.
- Supports 5000 pps rates of CCM traffic for bundle interfaces.
- Ethernet Connectivity Fault Management (CFM) is not supported with Maintenance association End Points (MEPs) that are configured on default and untagged encapsulated sub-interfaces that are part of a single physical interface.
- Multiple MEPs of different directions are not supported on the same interface or Xconnect.
- CFM does not support fast failover, which may result in session flaps on bundle interfaces. Use offload for virtual interfaces to avoid flaps on faster CCM intervals.

Ethernet SLA Statistics Measurement in a Profile

Table 8: Feature History Table

Feature Name	Release Information	Feature Description
Enhancement to Ethernet SLA Statistics Measurement	Release 7.7.1	You can now configure the size of bins for the delay and jitter measurement in Ethernet SLA statistics with a width value ranging from 1 to 10000000 microseconds. This enhancement provides granularity to store more accurate results of SLA statistics in the aggregate bins. In earlier releases, you could only configure the width value for the delay and jitter measurement in milliseconds. This feature introduces the usec keyword in the aggregate command.

The Ethernet SLA feature supports measurement of one-way and two-way delay and jitter statistics, and one-way FLR statistics.

Ethernet SLA statistics measurement for network performance is performed by sending packets and storing data metrics such as:

- Round-trip delay time—The time for a packet to travel from source to destination and back to source again.
- Round-trip jitter—The variance in round-trip delay time (latency).
- One-way delay and jitter—The router also supports measurement of one-way delay or jitter from source to destination, or from destination to source.
- One-way frame loss—The router also supports measurement of one-way frame loss from source to destination, or from destination to source.

In addition to these metrics, these statistics are also kept for SLA probe packets:

- Packet loss count
- Packet corruption event
- Out-of-order event
- Frame Loss Ratio (FLR)

Counters for packet loss, corruption, and, out-of-order packets are kept for each bucket, and in each case, a percentage of the total number of samples for that bucket is reported (for example, 4% packet corruption).

For delay, jitter, and loss statistics, the minimum, maximum, mean and standard deviation for the whole bucket are reported, as well as the individual samples or aggregated bins. Also, the overall FLR for the bucket, and individual FLR measurements or aggregated bins are reported for synthetic loss measurement statistics. The packet loss count is the overall number of measurement packets lost in either direction and the one-way FLR measures the loss in each direction separately.

When aggregation is enabled using the **aggregate** command, bins are created to store a count of the samples that fall within a certain value range, which is set by the **width** keyword. Only a counter of the number of results that fall within the range for each bin is stored. This uses less memory than storing individual results. When aggregation is not used, each sample is stored separately, which can provide a more accurate statistics analysis for the operation, but it is highly memory-intensive due to the independent storage of each sample.

A bucket represents a time period during which statistics are collected. All the results received during that time period are recorded in the corresponding bucket. If aggregation is enabled, each bucket has its own set of bins and counters, and only results relating to the measurements initiated during the time period represented by the bucket are included in those counters.

Frame Loss Ratio (FLR) is a primary attribute that can be calculated based on loss measurements. FLR is defined by the ratio of lost packets to sent packets and expressed as a percentage value. FLR is measured in each direction (source to destination and destination to source) separately. Availability is an attribute that is typically measured over a long period of time, such as weeks or months. The intent is to measure the proportion of time when there was prolonged high loss.

To configure one-way delay or jitter measurements, you must first configure the **profile (SLA)** command using the **type cfm-delay-measurement** form of the command.

For valid one-way delay results, you must have both local and remote devices time synchronized. In order to do this, you must select sources for frequency and time-of-day (ToD).

Frequency selection can be between any source of frequency available to the router, such as: BITS, GPS, SyncE, or PTP. The ToD selection is between the source selected for frequency and PTP or DTI. Note that NTP is not sufficient.

Configuration Guidelines



Caution Certain SLA configurations can use a large amount of memory which can affect the performance of other features on the router.

Before you configure Ethernet SLA, consider the following guidelines:

- Aggregation—Use of the **aggregate none** command significantly increases the amount of memory required because each individual measurement is recorded, rather than just counts for each aggregation bin. When you configure aggregation, consider that more bins will require more memory.
- Buckets archive—When you configure the **buckets archive** command, consider that the more history that is kept, the more memory will be used.
- Measuring two statistics (such as both delay and jitter) will use approximately twice as much memory as measuring one.
- Separate statistics are stored for one-way source-to-destination and destination-to-source measurements, which consumes twice as much memory as storing a single set of round-trip statistics.

- You must define the schedule before you configure SLA probe parameters to send probes for a particular profile. It is recommended to set up the profile—probe, statistics, and schedule before any commit.

Restrictions

One-way delay and jitter measurements are not supported by cfm-loopback profile types.

Configure Ethernet SLA Statistics Measurement in a Profile

To configure SLA statistics measurement in a profile, perform these steps:

1. Enter the Ethernet SLA configuration mode, using the **ethernet sla** command in Global Configuration mode.
2. Create an SLA operation profile with the **profile profile-name type cfm-delay-measurement** command.
3. Enable the collection of SLA statistics using the **statistics measure {one-way-delay-ds | one-way-delay-sd | one-way-jitter-ds | one-way-jitter-sd | round-trip-delay | round-trip-jitter | one-way-loss-ds | one-way-loss-sd}** command.
4. Configure the size and number of bins into which to aggregate the results of statistics collection. For delay measurements and data loss measurements, the default is that all values are aggregated into 1 bin. For synthetic loss measurements, by default the aggregation is disabled. Use the **aggregate {bins count width [usec] width | none}** command to configure the bins.
 - For delay and jitter measurements, you can configure a width value from 1 to 10000 milliseconds, if the number of bins is at least 2. To configure the width value in microseconds, use the **usec** option. You can configure the width value from 1 to 10000000 microseconds.
 - For data loss and synthetic loss measurements, you can configure a width value from 1 to 100 percentage points, if the number of bins is at least 2.
5. Configure the size of the buckets in which statistics are collected, using the **buckets size number probes** command.
6. Configure the number of buckets to store in memory using the **buckets archive number** command.
7. Save the configuration changes using the **end** or **commit** command.

Configuration Example

This example shows configuration of round-trip-delay statistics measurement in 5 bins each with a range of 123 microseconds:

```
Router(config)# ethernet sla
Router(config-sla)# profile test type cfm-delay-measurement
Router(config-sla-prof)# statistics measure round-trip-delay
Router(config-sla-prof-stat-cfg)# aggregate bins 5 width usec 123
Router(config-sla-prof-stat-cfg)# buckets size 1 probes
Router(config-sla-prof-stat-cfg)# buckets archive 50
Router(config-sla-prof-stat-cfg)# commit
```

This example shows configuration of round-trip-delay statistics measurement in 5 bins each with a range of 10 milliseconds:

```

Router(config)# ethernet sla
Router(config-sla)# profile test type cfm-delay-measurement
Router(config-sla-prof)# statistics measure round-trip-delay
Router(config-sla-prof-stat-cfg)# aggregate bins 5 width 10
Router(config-sla-prof-stat-cfg)# buckets size 1 probes
Router(config-sla-prof-stat-cfg)# buckets archive 50
Router(config-sla-prof-stat-cfg)# commit

```

Verification

This example displays aggregate bins configured with a range of 123 microseconds:

```

Router# show ethernet sla statistics detail
Tue Sep 28 07:59:22.340 PDT
Source: Interface GigabitEthernet0/0/0/2, Domain dom1
Destination: Target MAC Address 0012.0034.0056
=====
Profile 'test', packet type 'cfm-delay-measurement'
Scheduled to run every 1min first at 00:00:31 UTC for 10s

Round Trip Delay
~~~~~
1 probes per bucket

No stateful thresholds.

Bucket started at 07:56:31 PDT Tue 28 September 2021 lasting 10s
  Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
    Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
  Result count: 10
  Min: 0.000ms, occurred at 07:56:32 PDT Tue 28 September 2021
  Max: 1.000ms, occurred at 07:56:31 PDT Tue 28 September 2021
  Mean: 0.100ms; StdDev: 0.300ms

  Bins:
  Range                Samples    Cum. Count    Mean
  -----
    0 to 0.123 ms      9 (90.0%)    9 (90.0%)    0.000ms
    0.123 to 0.246 ms  0 (0.0%)    9 (90.0%)    -
    0.246 to 0.369 ms  0 (0.0%)    9 (90.0%)    -
    0.369 to 0.492 ms  0 (0.0%)    9 (90.0%)    -
    > 0.492 ms         1 (10.0%)   10 (100.0%)  1.000ms

```

This example displays aggregate bins configured with a range of 10 milliseconds:

```

Router# show ethernet sla statistics detail
Tue Sep 28 08:00:57.527 PDT
Source: Interface GigabitEthernet0/0/0/2, Domain dom1
Destination: Target MAC Address 0012.0034.0056
=====
Profile 'test', packet type 'cfm-delay-measurement'
Scheduled to run every 1min first at 00:00:31 UTC for 10s

Round Trip Delay
~~~~~
1 probes per bucket

No stateful thresholds.

Bucket started at 08:00:32 PDT Tue 28 September 2021 lasting 10s
  Pkts sent: 9; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
    Misordered: 1 (11.1%); Duplicates: 0 (0.0%)
  Result count: 9

```

Min: 0.000ms, occurred at 08:00:32 PDT Tue 28 September 2021
 Max: 0.000ms, occurred at 08:00:32 PDT Tue 28 September 2021
 Mean: 0.000ms; StdDev: 0.000ms

Results suspect due to a probe starting mid-way through a bucket

Bins: Range	Samples	Cum. Count	Mean
0 to 10 ms	9 (100.0%)	9 (100.0%)	0.000ms
10 to 20 ms	0 (0.0%)	9 (100.0%)	-
20 to 30 ms	0 (0.0%)	9 (100.0%)	-
30 to 40 ms	0 (0.0%)	9 (100.0%)	-
> 40 ms	0 (0.0%)	9 (100.0%)	-

Ethernet frame delay measurement for L2VPN services

Ethernet frame delay measurement complies with the ITU-T Y.1731 standard, which provides comprehensive fault management and performance monitoring recommendations. Delay Measurement Message (DMM) and Delay Measurement Reply (DMR) are used to periodically measure one-way or two-way frame delay and frame delay variation between a pair of point-to-point MEPs. Measurements are made between two MEPs belonging to the same domain and Maintenance Association (MA).

Table 9: Feature History Table

Feature Name	Release Information	Feature Description
Implement IGMP snooping	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) This feature is now supported on: • 8011-4G24Y4H-I
Ethernet frame delay measurement for L2VPN services	Release 24.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100]) (select variants only*) * This feature is supported on Cisco 8712-MOD-M routers.

Feature Name	Release Information	Feature Description
Ethernet frame delay measurement for L2VPN services	Release 7.5.3	You can now monitor L2VPN networks and avoid impact to your customers' operations by accurately measuring frame round-trip delays and jitters between two maintenance endpoints (MEPs). This feature lets you detect end-to-end connectivity, loopback, and link trace on MEPs. It reports service performance to your end customers, helping improve technical and operational tasks such as troubleshooting and billing. This feature introduces the cfm-delay-measurement probe command.

You can measure frame delay in the Layer 2 networks to detect end-to-end connectivity, loopback, and link trace on Maintenance End Points (MEPs) and also report service performance that helps to improve technical and operational tasks such as troubleshooting, billing, and so on. Frame delay is the duration between the time the source node transmits the first bit of a frame and the time the same source node receives the last bit of the frame.

The frame delay measurement uses the following two protocol data units (PDUs):

- Delay Measurement Message (DMM)—DMM is used to measure frame delay and frame delay variation between a pair of point-to-point Maintenance End Points (MEPs).
- Delay Measurement Response (DMR)—DMR is the delay measurement response sent by the destination MEP. When an MEP receives a DMM frame, the responder MEP responds with a DMR frame. The DMR frame carries a reply information and a copy of the timestamp contained in the DMM frame.



Note DMM sessions (using CFM) are not supported with MACsec enabled on the core interface, as this requires pre-encryption timestamping in the interface group.

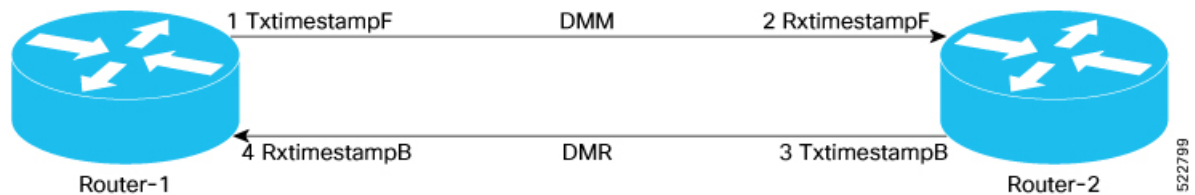
We support one-way and two-way frame delay measurement.

Frame Delay Measurement	Description
One-way frame delay measurement (1DM)	- Measures the frame delay on a unidirectional link between the MEPs. 1DM requires that clocks at both the transmitting MEP and the receiving MEPs are synchronized. - Measuring frame-delay variation does not require clock synchronization and the variation can be measured using 1DM and DMR frame combination.
Two-way frame delay measurement	- Measures the frame delay on a bidirectional link between the MEPs. - Two-way delay measurement does not require the clocks at both the transmitting MEP and the receiving MEPs to be synchronized. - The two-way frame delay is measured using only DMM and DMR frames.

For more information about CFM, see [Configuring Ethernet OAM, on page 1](#).

Topology

Let's see how a round-trip frame delay is measured with the following sample topology.



- The sender MEP (Router-1) transmits a frame containing delay measurement request information and the timestamp at the which router sends the DMM.
- When packets pass through each interface, timestamps are written into DMMs and DMRs at both local and peer MEPs.
- When the DMM leaves the local interface, the TX timestamp is added to the packet.
- When the receiver MEP (Router-2) receives the frame, records the timestamp at which the receiver MEP receives the frame with the delay measurement request information and the remote MEP (Router-2) responds with an DMR adding the remote TX timestamp to the packet as it leaves the remote interface.

To measure a round-trip delay for a traffic exchange between Router-1 and Router-2, four timestamps get populated as the packet moves through the network.

- Router-1 adds the TxTimestampF when DMM packet is transmitted.
- Router-2 adds RxTimestampF when DMM packet is received by it.

- Router-2 adds TxTimestampB when DMR packet is transmitted.
- Router-1 adds RxTimestampB when DMR is received by it

The round-trip delay is calculated using the following formula:

$$\begin{aligned} \text{Delay} &= (\text{RxTimestampB} - \text{TxTimestampF}) - (\text{TxTimestampB} - \text{RxTimestampF}) \\ &= \text{RxTimestampB} - \text{TxTimestampF} - \text{TxTimestampB} + \text{RxTimestampF} \\ &= (\text{RxTimestampF} - \text{TxTimestampF}) - (\text{TxTimestampB} - \text{RxTimestampB}) \end{aligned}$$

Configure Ethernet Frame Delay Measurement for L2VPN Services

Perform the following tasks to configure Ethernet Frame Delay Measurement for L2VPN Services:

1. Configure L2VPN service.
2. Enable CFM service continuity check.
3. Enable CFM on the interface.
4. Configure Ethernet frame delay measurement.

```
/* Configure L2VPN service */
Router# configure
Router(config)# l2vpn
Router(config-l2vpn)# xconnect group evpn_vpws_203
Router(config-l2vpn-xc)# p2p evpn_vpws_phy-100
Router(config-l2vpn-xc-p2p)# interface GigabitEthernet0/0/0/2.100
Router(config-l2vpn-xc-p2p)# neighbor evpn evi 30001 target 30001 source 50001
Router(config-l2vpn-xc-p2p)# commit

/* Enable CFM service continuity check */
Router# ethernet cfm
Router(config-cfm# domain xcup1 level 7 id null
Router(config-cfm-dmn)# service xcup1 xconnect group evpn_vpws_Bund
Router(config-cfm-dmn-svc)# mip auto-create all ccm-learning
Router(config-cfm-dmn-svc)# continuity-check interval 1s
Router(config-cfm-dmn-svc)# mep crosscheck
Router(config-cfm-dmn-svc)# mep-id 4001
Router(config-cfm-dmn-svc)# commit

/* Enable CFM on the interface */
Router(config)# interface GigabitEthernet0/0/0/2.100 l2transport
Router(config-subif)# encapsulation dot1q 100
Router(config-subif)# rewrite ingress tag pop 1 symmetric
Router(config-subif)# mtu 9100
Router(config-subif)# ethernet cfm
Router(config-if-cfm)# mep domain bd-domain service bd-service mep-id 4001
Router(config-if-cfm-mep)# sla operation profile test-profile1 target mep-id 1112
Router(config-if-cfm-mep)# commit

/* Configure Ethernet frame delay measurement */
Router(config)# ethernet sla
Router(config-sla)# profile EVC-1 type cfm-delay-measurement
Router(config-sla-prof)# probe
Router(config-sla-prof-pb)# send packet every 1 seconds
Router(config-sla-prof-pb)# schedule
Router(config-sla-prof-schedule)# every 3 minutes for 120 seconds
Router(config-sla-prof-schedule)# statistics
Router(config-sla-prof-stat)# measure round-trip-delay
```

```
Router(config-sla-prof-stat-cfg)# buckets size 1 probes
Router(config-sla-prof-stat-cfg)# buckets archive 5
Router(config-sla-prof-stat-cfg)# commit
```

Running Configuration

This section shows the Ethernet frame delay measurement running configuration.

```
/* Configure L2VPN service */
l2vpn
xconnect group evpn_vpws_203
p2p evpn_vpws_phy-100
interface GigabitEthernet0/0/0/2.100
neighbor evpn evi 30001 target 30001 source 50001
!
/* Enable CFM service continuity check */
ethernet cfm
domain xcup1 level 7 id null
service xcup1 xconnect group evpn_vpws_Bundle_ether203 p2p evpn_vpws-100 id number 4001
mip auto-create all ccm-learning
continuity-check interval 1s
mep crosscheck
mep-id 4001
!
/* Enable CFM on the interface */
interface GigabitEthernet0/0/0/2.100 l2transport
encapsulation dot1q 100
rewrite ingress tag pop 1 symmetric
mtu 9100
ethernet cfm
mep domain bd-domain service bd-service mep-id 4001
sla operation profile test-profile1 target mep-id 1112
!
/* Configure Ethernet SLA */
ethernet sla
profile EVC-1 type cfm-delay-measurement
probe
send packet every 1 seconds
!
schedule
every 3 minutes for 120 seconds
!
statistics
measure round-trip-delay
buckets size 1 probes
buckets archive 5
!
```

Verification

Verify the frame delay measurement. In the following example, you observe that the sent and received DMM and DMR packets are same. So there is no delay in frame transimission.

```
Router# show ethernet cfm local meps interface GigabitEthernet0/0/0/2.100 verbose

Up MEP on GigabitEthernet0/0/0/2.100 MEP-ID 4001
=====
Interface state: Up      MAC address: 0c11.6752.3af8
Peer MEPs: 1 up, 0 with errors, 0 timed out (archived)

CCM generation enabled: Yes, 10s (Remote Defect detected: No)
```

```

AIS generation enabled: No
Sending AIS:           No
Receiving AIS:         No
Sending CSF:           No
Receiving CSF:         No

```

```

Packet      Sent      Received
-----
CCM          19        9 (out of seq: 0)
DMM          473        0
DMR           0        473

```

Minimum delay bin

Y.1731 Ethernet SLA is used to collect and optionally aggregate performance metrics over time. The data collected during a specific timeframe is organized into buckets. When aggregation is enabled, each bucket contains a set of bins, which helps reduce memory consumption.

With the minimum-delay bin feature, you can configure a distinct width for the first bin. This prevents the wastage of bins that might otherwise be empty due to the inherent speed of light delays. The remaining bins can then focus on capturing variations in observed delays.

Table 10: Feature History Table

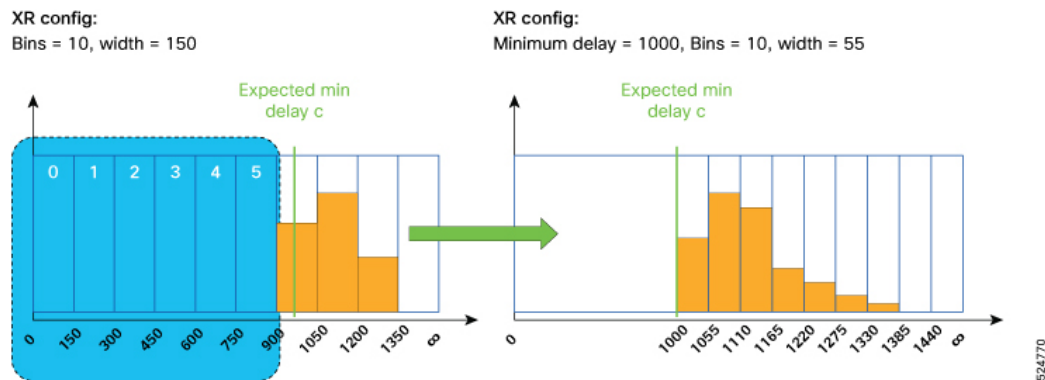
Feature Name	Release Information	Feature Description
Minimum-delay bin	Release 25.1.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q200, P100], 8700 [ASIC: P100, K100]); Centralized Systems (8600 [ASIC: Q200]); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]). For statistics aggregation, you can now configure a distinct width for the first bin to adjust for large propagation delay. By using this feature, you can avoid wasting several bins that would be empty in some unavoidable situations such as delay due to speed of light limitations. The feature introduces these changes: CLI: - The minimum-delay keyword is introduced in the aggregate and cfm-delay-measurement probe commands. YANG Data Models: New XPaths for - Cisco-IOS-XR-infra-sla-cfg.yang - Cisco-IOS-XR-um-ethernet-sla-cfg.yang - Cisco-IOS-XR-infra-sla-oper.yang (see GitHub, YANG Data Models Navigator)

Information about minimum-delay bin

In situations where a delay is expected in first few iterations, you can specify the width of the first bin independently using the **minimum-delay** keyword in statistics aggregation configuration. See below examples to configure aggregation with and without minimum-delay support.

The below figure shows the comparison between statistics aggregation before and after minimum-delay configuration.

Figure 8: Comparison diagram before and after minimum-delay configuration



Example 1: Configuring aggregation without minimum-delay

```
statistics measure round-trip-delay aggregate bins 10 width 150
```

This configuration has 10 bins and the results being aggregated into the ranges 0-150, 150-300, 300-450 and so on, until 1350+, where the last bin has infinite width to hold all values greater than 1350.

Example 2: Configuring aggregation with minimum-delay

```
statistics measure round-trip-delay aggregate bins 10 width 55 minimum-delay 1000
```

Here, the width of the first bin is 1000ms and not 150ms. The width of the other nine bins are aggregated into 1000-1055, 1055-1110, 1110-1165, and so on. This leads to increased resolution with the same number of bins, as all the bins are utilized.



Note To specify the values of width and minimum-delay in microseconds instead of milliseconds, you must use the **usec** keyword. For more information, see [aggregate](#) and [cfm-delay-measurement probe](#) commands.

Configure minimum delay bin support

Before you begin

Make sure that the number of bins for aggregate configuration is at least two.

Procedure

Step 1 Configure an SLA Operation profile and statistics measurement for the Profile.

Example:

Configure Ethernet Frame Delay Measurement for L2VPN Services.

```
Router(config)# ethernet sla
Router(config-sla)# profile EVC-1 type cfm-delay-measurement
Router(config-sla-prof)# probe
Router(config-sla-prof-pb)# send packet every 1 seconds
Router(config-sla-prof-pb)# schedule
Router(config-sla-prof-schedule)# every 3 minutes for 120 seconds
Router(config-sla-prof-schedule)# statistics
Router(config-sla-prof-stat)# measure round-trip-delay
Router(config-sla-prof-stat-cfg)# buckets size 1 probes
Router(config-sla-prof-stat-cfg)# buckets archive 5
Router(config-sla-prof-stat-cfg)# commit
```

- Step 2** Configure aggregation for the SLA profile and then configure the width of the first bin by using the **minimum-delay** keyword.

Example:

Configure aggregation with bin count of 10 and the width of the first bin as 5 ms.

```
Router(config-sla-prof-schedule)# statistics
Router(config-sla-prof-stat)# measure round-trip-delay
Router(config-sla-prof-stat-cfg)# aggregate bins 5 width 10 minimum-delay 30
Router(config-sla-prof-stat-cfg)# buckets size 1 probes
Router(config-sla-prof-stat-cfg)# buckets archive 5
Router(config-sla-prof-stat-cfg)# commit
```

- Step 3** View the running configuration using the **show running-config** command.

Example:

```
ethernet sla
profile EVC-1 type cfm-delay-measurement
  probe
    send packet every 1 seconds
  !
  schedule
    every 3 minutes for 120 seconds
  !
  statistics
    measure round-trip-delay
    aggregate bins 10 width 2 minimum-delay 5
    buckets size 1 probes
    buckets archive 5
  !
```

- Step 4** Verification

Verify the output by using the below show commands.

- **show protocolsla operations detail**
- **show protocolsla probes**
- **show protocolsla statistics**

Example:

Use the below show command to verify the output.

```
router# show ethernet sla statistics history detail on-demand
```

Below is a sample output.

Bucket started at 15:38 on Tue 02 Jul 2024, lasting 1 hour:

Pkts sent: 1200; ...

Result count: 30

Min: 13ms; Max: 154ms; Mean: 28ms; StdDev: 11ms

Bins:

Range	Samples	Cum. Count	Mean
0 to 30 ms	20 (2%)	20 (2%)	37ms
30 to 35 ms	909 (61%)	929 (77%)	67ms
35 to 40 ms	212 (18%)	1141 (95%)	75ms
40 to 45 ms	98 (11%)	1141 (95%)	75ms
> 45 ms	55 (5%)	1196 (85%)	90ms

Link loss forwarding

Link loss forwarding (LLF) is a mechanism used in networking to propagate the status of a network link to other connected devices. When a link experiences a failure or goes down, LLF ensures that this information is forwarded to other network devices, which can then take appropriate actions to maintain network stability and performance.

Table 11: Feature History Table

Feature Name	Release Information	Feature Description
Link loss forwarding	Release 25.1.1	Introduced in this release on: Fixed Systems (8010 [ASIC: A100]) This feature is now supported on: • 8011-4G24Y4H-I
Link loss forwarding	Release 24.4.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100]) (select variants only*) * This feature is supported on Cisco 8712-MOD-M routers.

Feature Name	Release Information	Feature Description
Link loss forwarding	Release 7.9.1	We have now enabled high availability between two bridged interfaces by disabling both interfaces if any one of them fails. Such high availability is enabled because the functionality allows a fault detected on one side of a CFM-protected network to propagate to the other, allowing the device to re-route around the failure. In earlier releases, a failure on one bridged interface did not disable the other interface, and connected devices remained unaware of the link loss. The feature introduces these changes: • CLI: New <code>propagate-remote-status</code> command • YANG Data Model: New XPaths for <code>Cisco-IOS-XR-um-ethernet-cfm-cfg.yang</code> (see GitHub, YANG Data Models Navigator)

You can enable LLF on a network by one of the following methods:

- **Link State Monitor and Propagation by CFM:** LLF uses Connectivity Fault Management (CFM) to transmit notification of a signal loss or fault across the network. When there is a fault on a link to a device on one side of the network, the connection to the port on the other side needs to be shutdown so that the device re-routes the traffic.
- **Remote Link State Propagation:** LLF uses this method for Layer 2 transport events to propagate link failures to remote endpoints. When a link failure occurs, LLF ensures that the failure is communicated to other devices in the network. This enables the other devices to take appropriate action, such as rerouting traffic or triggering failover mechanisms.

Link State Monitor and Propagation by CFM

Link State Monitoring involves tracking the status of network links to ensure they are operational and performing as expected. This can include monitoring for link failures, degradations, or other issues that might affect network performance. When a link state changes, this information needs to be propagated throughout the network so that other devices can adjust their routing tables and network operations accordingly.

When there is a fault on a link to a device on one side of the network, the connection to the port on the other side needs to be shutdown so that the device re-routes the traffic. This requires the interface to be TX-disabled.

Link Loss Forwarding (LLF) uses Connectivity Fault Management (CFM) to transmit notification of a signal loss or fault across the network. If a local attachment circuit (AC) on a bridged interface fails, one of the following signals or packet types are sent to the neighboring device:

- **Continuity Check Message (CCM)** – The CCMs are heartbeat messages exchanged periodically between all the Maintenance End Points (MEPs) in a service. MEPs are members of a particular service within a domain and are responsible for sourcing and sinking CFM frames. Each MEP sends out multicast CCMs, and receives CCMs from all the other MEPs in the service. This allows each MEP to discover its peer MEPs, and to verify that there is connectivity between them.

- Alarm Indication Signal (AIS) – These are messages sent periodically by MEPs that have detected a fault, to the MEPs in the next highest maintenance domain level.
- Client Signal Fail (CSF) – A mechanism for error detection. When a MEP detects an issue, the MEP sends CSF packets to its peer MEPs.

For more information on MEPs, see [Maintenance Points, on page 14](#).

Connectivity Fault Management Daemon (CFMD) and Ether-MA are processes that run on the control plane of the router. Ether-MA handles owner channel communication and resyncs from CFMD, L2VPN, and other Ether MA processes. This module handles the TX-disable and TX-enable events, based on the notifications from CFMD.

When the system receives a CCM or AIS with fault indication, or a CSF error packet, CFMD communicates with Ether-MA to TX-disable the interface.

When an interface receives a fault notification, the transitions are handled as follows:

- The interface is transitioned to TX-disable state.
- A restore or damping timer with a $3.5 * \text{packet interval duration}$ is started.
- If no other fault packets are received after the restore timer ends, the TX-disable state is cleared and the interface is transitioned to TX-enable state.

Restrictions for Link Loss Forwarding for CFM

- Link loss forwarding is not permitted on subinterfaces.
- Link loss forwarding is permitted only on UP MEPs. The UP MEPs send the frames into the bridge relay function and not through the wire connected to the port where the MEP is configured. For more information on UP MEPs, see [MEP and CFM Processing Overview, on page 14](#).
- A damping or restore timer governs transitions of an interface from TX-disabled state to TX-enabled state. The period of the damping timer is calculated by three times the configured CCM interval. You cannot configure the damping timer.
- The damping timer is not provided for transitions of an interface from TX-enabled state to TX-disabled state.
- Link loss forwarding does not work on bundle interfaces configured with LACP.

Configure Link Loss Forwarding for CFM

To configure LLF on a network:

1. Configure a Connectivity Fault Management (CFM) domain and service.
2. Configure a Maintenance End Point (MEP) under the CFM domain and service.
3. Configure continuity check message (CCM) interval on the MEP. The restore timer for a CCM notification is calculated based on the configured CCM interval.
4. Configure Client Signal Fail (CSF) transmission on the MEP, to enable CSF transmission.
5. Configure CSF logging on the MEP, to enable logging on receiving a CSF packet.



Note The CSF configuration is optional and is not required when both the devices in CFM-protected network are running with IOS-XR. This configuration is required for inter-operation with certain client-end setups that contain devices from other clients.

6. Enable LLF on an interface using the **propagate-remote-status** command. This command triggers the interface to be TX-disabled on fault detection.

Configuration Example

```
/* Configure CFM domain, service, and MEP */

Router# configure
Router(config)# ethernet cfm
Router(config-cfm)# domain dom1 level 1 service ser1 bridge group up-meps bridge-domain
up-mep

/* Configure CCM interval */

Router(config-cfm-dmn-svc)# continuity-check interval 1m

/* (Optional) Configure CSF */

Router(config-cfm-dmn-svc)# csf interval 1m cos 4
Router(config-cfm-dmn-svc)# csf-logging
Router(config-cfm-dmn-svc)# commit

/* Enable LLF on an interface */

Router# configure
Router(config)# interface GigabitEthernet0/2/0/0
Router(config-if)# ethernet cfm
Router(config-if-cfm)# mep domain dom1 service ser1 mep-id 1
Router(config-if-cfm-mep)# propagate-remote-status
Router(config-if-cfm-mep)# commit
```

Running Configuration

```
ethernet cfm
 domain dom1 level 1
   service ser1 bridge group up-meps bridge-domain up-mep
   continuity-check interval 1m
   csf interval 1m cos 4
   csf-logging
 !
!
interface GigabitEthernet0/2/0/0
 ethernet cfm
   mep domain dom1 service ser1 mep-id 1
   propagate-remote-status
 !
!
!
```

Verification

The following output shows LLF configuration and fault state for each interface:

```
Router# show ethernet cfm interfaces llf location 0/RP0/CPU0
Defects (from at least one peer MEP):
A - AIS received           I - Wrong interval
R - Remote Defect received V - Wrong Level
L - Loop (our MAC received) T - Timed out (archived)
C - Config (our ID received) M - Missing (cross-check)
X - Cross-connect (wrong MAID) U - Unexpected (cross-check)
P - Peer port down         F - CSF received
```

```
GigabitEthernet0/1/0/0
MEP Defects                                Restore Timer
-----
100 R                                     Not running
101 None                                10s remaining
102 RPF                                 Not running
```

```
GigabitEthernet0/1/0/1
MEP Defects                                Restore Timer
-----
110 None                                3s remaining
```

```
GigabitEthernet0/1/0/2
MEP Defects                                Restore Timer
-----
120 P                                     Not running
```

The following output shows that the interface received a single CSF packet at 1 minute interval, so that the interface is TX-disabled with a damping timer of 3.5 minutes.

```
Router# show ethernet cfm local meps detail
Domain dom1 (Level 1), Service ser1
UP MEP on GigabitEthernet0/1/0/0 MEP-ID 1
=====
Interface state: UP      MAC address: 0204.3dbe.c93b
Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)
Cross-check errors: 0 missing, 0 unexpected

CCM generation enabled: No
AIS generation enabled: No
Sending AIS:           No
Receiving AIS:         No
Sending CSF:           No
Receiving CSF:         Yes (Interval: 1min, started 00:03:29 ago)
TX Disable triggered:  Yes (restore timer not running)
```

The following output shows that the interface received a CCM notification that the peer MEP port is down, so that the interface is TX-disabled.

```
Router# show ethernet cfm local meps detail
Domain dom1 (Level 1), Service ser1
UP MEP on GigabitEthernet0/1/0/0 MEP-ID 1
=====
Interface state: UP      MAC address: 0204.3dbe.c93b
Peer MEPs: 1 up, 1 with errors, 0 timed out (archived)
Cross-check errors: 0 missing, 0 unexpected

CCM generation enabled: Yes, 1min (Remote Defect detected: Yes)
CCM defects detected:   P - peer port down
AIS generation enabled: No
Sending AIS:           No
Receiving AIS:         No
Sending CSF:           No
Receiving CSF:         No
TX Disable triggered:  Yes (restore timer not running)
```

The following output shows that the interface received CCM notification that the peer MEP port is up, and restore timer is started for the TX-disabled interface.

```
Router# show ethernet cfm local meps detail
Domain dom1 (Level 1), Service ser1
UP MEP on GigabitEthernet0/1/0/0 MEP-ID 1
=====
Interface state: UP      MAC address: 0204.3dbe.c93b
Peer MEPS: 1 up, 0 with errors, 0 timed out (archived)
Cross-check errors: 0 missing, 0 unexpected

CCM generation enabled: Yes, 1min (Remote Defect detected: No)
AIS generation enabled: No
Sending AIS:           No
Receiving AIS:         No
Sending CSF:           No
Receiving CSF:         No
TX Disable triggered:  Yes (restore timer running, 1183ms remaining)
```

The following output shows Ether-MA configured bundles and their members:

```
Router# show ethernet infra internal ether-ma bundles
Bundle interface: Bundle-Ether1 (TX disabled)
Bundle members:
  GigabitEthernet0/1/0/1
  GigabitEthernet0/1/0/2

Bundle interface: Bundle-Ether2
Bundle members:
  GigabitEthernet0/2/0/1
```

Remote link state propagation

Table 12: Feature History Table

Feature Name	Release Information	Feature Description
Remote link state propagation	Release 25.1.1	Introduced in this release on: Fixed Systems (8700 [ASIC: K100], 8010 [ASIC: A100]) This feature is now supported on: • 8712-MOD-M • 8011-4G24Y4H-I
Remote Link State Propagation	Release 24.4.1	Introduced in this release on: Fixed Systems(8200, 8700);Modular Systems (8800 [LC ASIC: P100]) (select variants only*) *This feature is now supported on: • 8212-32FH-M • 8711-32FH-M • 88-LC1-12TH24FH-E

Feature Name	Release Information	Feature Description
Remote link state propagation	Release 24.3.1	Introduced in this release on: Fixed Systems (8200 [ASIC: Q200, P100], 8700 [ASIC: P100]); Centralized Systems (8600 [ASIC:Q200]); Modular Systems (8800 [LC ASIC: Q100, Q200, P100]) Remote Link State Propagation allows the status of a link to be communicated to remote devices, ensuring that all relevant parts of the network are aware of link state changes. Link Loss Forwarding (LLF) uses this feature to propagate link failures to remote endpoints. By enabling remote state propagation and LLF on an interface, you can ensure that the link state changes are communicated to remote devices, allowing for quick failover and rerouting of traffic. This feature introduces the propagate remote-status command.

Remote Link State Propagation allows the status of a link to be communicated to remote devices, ensuring that all relevant parts of the network are aware of link state changes. This is particularly useful in Layer 2 transport networks, where maintaining accurate link status information is crucial for network performance and reliability.

Link Loss Forwarding for Layer 2 Transport

Link Loss Forwarding (LLF) uses Remote Link State Propagation to propagate link failures to remote endpoints. When a link failure occurs, LLF ensures that the failure is communicated to other devices in the network, allowing them to take appropriate action, such as rerouting traffic or triggering failover mechanisms. LLF helps avoid packet loss and triggers network convergence through alternate links. It works by sending signals across the pseudowire (PW) to the neighboring device, bringing the PW and far-end attachment circuit (AC) down if the local AC goes down.

You can configure LLF for the Layer 2 transport events, using the **propagate remote-status** command. You can enable LLF on the following interface types: 1G, 10G, 25G, 40G, 100G, and 400G.

Remote State Propagation with LLF for L2 Transport

When you configure LLF for L2 transport events, the following are the processes that happen:

1. **Link State Detection:** The network device detects a change in the link state, such as a link going down.

2. **Remote State Propagation:** When remote state propagation is enabled, the detected link state change is propagated to remote devices. This ensures that all relevant devices are aware of the link failure and can take appropriate action.
3. **LLF Activation:** LLF uses the propagated link state information to trigger failover mechanisms or reroute traffic. This helps maintain network performance and reliability by quickly responding to link failures.

When you enable remote state propagation and LLF on an interface, the link state changes are communicated to remote devices allowing for quick failover and rerouting of traffic.

Configure Link Loss Forwarding for Layer 2 Transport

The following example shows how to configure LLF for Layer 2 transport events.

Procedure

- Step 1** To enable link loss forwarding for Layer transport events, enter the interface mode, configure **l2transport**, and then enable LLF using the **propagate remote-status** command.

```
Router(config)# interface tenGigE 0/0/0/1
Router(config-if)# l2transport
Router(config-if-l2)# propagate remote-status
Router(config-if-l2)# commit
```

- Step 2** View the running configuration.

```
interface TenGigE 0/0/0/1
  l2transport
  propagate remote-status
!
```

