Configuring Ethernet OAM

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

Feature History for Configuring Ethernet OAM

Release	Modification
Release 6.1.1	Support for the following features was introduced: • Ethernet Link OAM • Ethernet CFM
Release 7.1.1	Support for CFM adaptive bandwidth notifications was introduced for NCS5500 platforms.
Release 7.5.1	Support for Link Loss Forwarding (LLF) was introduced.
Release 7.5.1	Support for CFM adaptive bandwidth notifications was introduced for Cisco Network Convergence System 5700 Series routers and routers with Cisco NC57 line cards operating in native mode.

Ethernet OAM

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

Ethernet Link OAM

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 . 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.

When an EOAM packet is received on any one of the AC interfaces on which EOAM is not configured, the AC interface multicasts the received EOAM packets to other AC interfaces that are part of EVPN-BD to reach the peer. When an EOAM is enabled on the bundle member in the peer, it punts the packet to the CPU in the peer. Also, the EOAM flaps the bundle member as the local or remote Key of the received EOAM does not match.

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, such as CFM, to control the `line protocol` state of an interface.

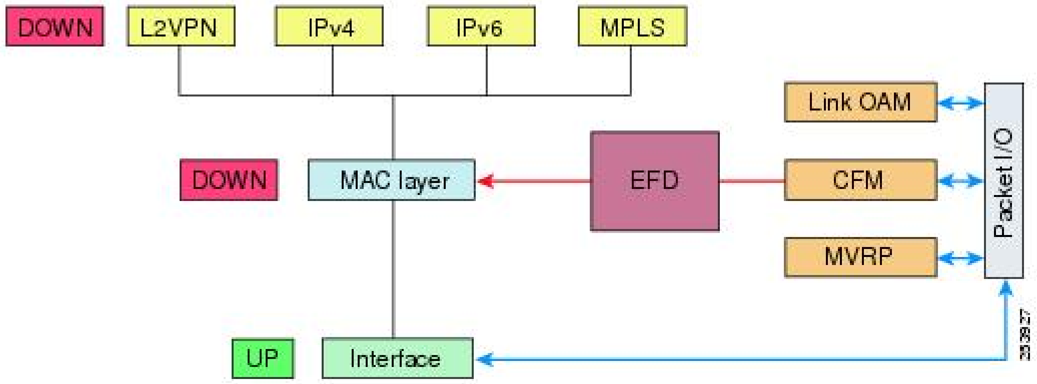
Unlike many other interface types, Ethernet interfaces do not have a line protocol, whose state is independent from that of the interface. For Ethernet interfaces, this role is handled by the physical-layer Ethernet protocol itself, and therefore if the interface is physically up, then it is available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

EFD changes this to allow CFM to act as the line protocol for Ethernet interfaces. This allows CFM to control the interface state so that if a CFM 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 CFM frames continue to flow. This allows CFM to detect when the problem has been resolved, and thus bring the interface backup automatically.

This figure shows CFM 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 all higher level protocols (Layer 2 pseudowires, IP protocols, and so on) to go down and also trigger a reconvergence where possible. As soon as CFM detects there is no longer any error, it can signal to EFD and all protocols will once again go active.

Figure 1: CFM Error Detection and EFD Trigger



MIB Retrieval

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 (Cisco-Proprietary)

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

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

How to Configure Ethernet OAM

This section provides these configuration procedures:

Configuring Ethernet Link 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.

SUMMARY STEPS

1. **configure**
2. **ethernet oam profile** *profile-name*
3. **link-monitor**
4. **symbol-period window** *window*
5. **symbol-period threshold low** *threshold* **high** *threshold*
6. **frame window** *window*
7. **frame threshold low** *threshold* **high** *threshold*
8. **frame-period window** *window*
9. **frame-period threshold low***threshold* **high** *threshold*
10. **frame-seconds window** *window*
11. **frame-seconds threshold low** *threshold* **high** *threshold*
12. **exit**
13. **mib-retrieval**
14. **connection timeout** <*timeout*>
15. **hello-interval** {**100ms**|**1s**}
16. **mode** {**active**|**passive**}
17. **require-remote mode** {**active**|**passive**}
18. **require-remote mib-retrieval**
19. **action capabilities-conflict** {**disable | efd | error-disable-interface**}
20. **action critical-event** {**disable | error-disable-interface**}
21. **action discovery-timeout** {**disable | efd | error-disable-interface**}
22. **action dying-gasp** {**disable | error-disable-interface**}
23. **action high-threshold** {**error-disable-interface | log**}
24. **action session-down** {**disable | efd | error-disable-interface**}
25. **action session-up disable**
26. **action uni-directional link-fault** {**disable | efd | error-disable-interface**}
27. **action wiring-conflict** {**disable | efd | log**}
28. **uni-directional link-fault detection**

29. **commit**
30. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure terminal`	Enters global configuration mode.
Step 2	**ethernet oam profile** *profile-name* **Example:** `RP/0/RP0/CPU0:router(config)# ethernet oam profile Profile_1`	Creates a new Ethernet Operations, Administration and Maintenance (OAM) profile and enters Ethernet OAM configuration mode.
Step 3	**link-monitor** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# link-monitor`	Enters the Ethernet OAM link monitor configuration mode.
Step 4	**symbol-period window** *window* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# symbol-period window 60000`	(Optional) Configures the window size (in milliseconds) for an Ethernet OAM symbol-period error event. The IEEE 802.3 standard defines the window size as a number of symbols rather than a time duration. These two formats can be converted either way by using a knowledge of the interface speed and encoding. The range is 1000 to 60000. The default value is 1000.
Step 5	**symbol-period threshold low** *threshold* **high** *threshold* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# symbol-period threshold ppm low 1 high 1000000`	(Optional) Configures the thresholds (in symbols) that trigger an Ethernet OAM symbol-period error event. The high threshold is optional and is configurable only in conjunction with the low threshold. The range is 0 to 60000000. The default low threshold is 1.
Step 6	**frame window** *window* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame window 6000`	(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:**	(Optional) Configures the thresholds (in symbols) that triggers an Ethernet OAM frame error event. The high threshold is optional and is configurable only in conjunction with the low threshold.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-eoam-lm)# frame threshold low 10000000 high 60000000`	The range is from 0 to 60000000. The default low threshold is 1.

Step 8	**frame-period window** *window*
	Example:
	`RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period window 60000`
	`RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period window milliseconds 60000`
		(Optional) Configures the window size (in milliseconds) for an Ethernet OAM frame-period error event. The IEEE 802.3 standard defines the window size as number of frames rather than a time duration. These two formats can be converted either way by using a knowledge of the interface speed. Note that the conversion assumes that all frames are of the minimum size.
		The range is from 1000 to 60000.
		The default value is 1000.
		Note The only accepted values are multiples of the line cardinterface module-specific polling interval, that is, 1000 milliseconds for most line cardsinterface modules.

Step 9	**frame-period threshold low***threshold* **high** *threshold* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame-period threshold ppm low 100 high 1000000`	(Optional) Configures the thresholds (in errors per million frames) that trigger an Ethernet OAM frame-period error event. The frame period window is defined in the IEEE specification as a number of received frames, in our implementation it is x milliseconds. The high threshold is optional and is configurable only in conjunction with the low threshold. The range is from 1 to 1000000. The default low threshold is 1. To obtain the number of frames, the configured time interval is converted to a window size in frames using the interface speed. For example, for a 1Gbps interface, the IEEE defines minimum frame size as 512 bits. So, we get a maximum of approximately 1.5 million frames per second. If the window size is configured to be 8 seconds (8000ms) then this would give us a Window of 12 million frames in the specification's definition of Errored Frame Window. The thresholds for frame-period are measured in errors per million frames. Hence, if you configure a window of 8000ms (that is a window of 12 million frames) and a high threshold of 100, then the threshold would be crossed if there are 1200 errored frames in that period (that is, 100 per million for 12 million).
Step 10	**frame-seconds window** *window* **Example:**	(Optional) Configures the window size (in milliseconds) for the OAM frame-seconds error event. The range is 10000 to 900000.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds window 900000`	The default value is 60000. **Note** The only accepted values are multiples of the line cardinterface module-specific polling interval, that is, 1000 milliseconds for most line cardsinterface modules.
Step 11	**frame-seconds threshold low** *threshold* **high** *threshold* **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# frame-seconds threshold low 3 high 900`	(Optional) Configures the thresholds (in seconds) that trigger a frame-seconds error event. The high threshold value can be configured only in conjunction with the low threshold value. The range is 1 to 900 The default value is 1.
Step 12	**exit** **Example:** `RP/0/RP0/CPU0:router(config-eoam-lm)# exit`	Exits back to Ethernet OAM mode.
Step 13	**mib-retrieval** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# mib-retrieval`	Enables MIB retrieval in an Ethernet OAM profile or on an Ethernet OAM interface.
Step 14	**connection timeout** <*timeout*> **Example:** `RP/0/RP0/CPU0:router(config-eoam)# connection timeout 30`	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** {**100ms**|**1s**}
	Example:
	`RP/0/RP0/CPU0:router(config-eoam)# hello-interval 100ms`
		Configures the time interval between hello packets for an Ethernet OAM session. The default is 1 second (**1s**).

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.

	Command or Action	Purpose
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**}
	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.
		Note • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.

Step 20	**action critical-event** {**disable** | **error-disable-interface**}
	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.
		Note • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.

Step 21	**action discovery-timeout** {**disable** | **efd** | **error-disable-interface**}
	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.
		Note • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.

Step 22	**action dying-gasp** {**disable** | **error-disable-interface**}
	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.
		Note • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.

	Command or Action	Purpose

Step 23	**action high-threshold** {**error-disable-interface** | **log**}
	Example:
	`RP/0/RP0/CPU0:router(config-eoam)# action high-threshold error-disable-interface`
		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.
		Note • If you change the default, the **disable** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and take no action at the interface when the event occurs.

Step 24	**action session-down** {**disable** | **efd** | **error-disable-interface**}
	Example:
	`RP/0/RP0/CPU0:router(config-eoam)# action session-down efd`
		Specifies the action that is taken on an interface when an Ethernet OAM session goes down.
		Note • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.

Step 25	**action session-up disable** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# action session-up disable`	Specifies that no action is taken on an interface when an Ethernet OAM session is established. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.
Step 26	**action uni-directional link-fault** {**disable** | **efd** | **error-disable-interface**}	Specifies the action that is taken on an interface when a link-fault notification is received from the remote Ethernet OAM peer. The default action is to create a syslog entry. **Note** • If you change the default, the **log** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and log the event for the interface when it occurs.

Step 27	**action wiring-conflict** {**disable** | **efd** | **log**}
	Example:
	`RP/0/RP0/CPU0:router(config-eoam)# action session-down efd`
		Specifies the action that is taken on an interface when a wiring-conflict event occurs. The default is to put the interface into error-disable state.
		Note • If you change the default, the **error-disable-interface** keyword option is available in Interface Ethernet OAM configuration mode to override the profile setting and put the interface into error-disable state when the event occurs.

	Command or Action	Purpose
Step 28	**uni-directional link-fault detection** **Example:** `RP/0/RP0/CPU0:router(config-eoam)# uni-directional link-fault detection`	Enables detection of a local, unidirectional link fault and sends notification of that fault to an Ethernet OAM peer.
Step 29	**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 30	**end**
	Example:
	`RP/0/RP0/CPU0:router(config-if)# end`
		Ends the configuration session and exits to the EXEC mode.

Attaching an Ethernet OAM Profile to an Interface

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

SUMMARY STEPS

1. **configure**
2. **interface** [**FastEthernet** | **HundredGigE**| **TenGigE**] *interface-path-id*
3. **ethernet oam**
4. **profile** *profile-name*
5. **commit**
6. **end**

DETAILED STEPS

	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**] *interface-path-id* **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 *rack/slot/module/port*. **Note** • The example indicates an 8-port 10-Gigabit Ethernet interface in modular services card slot 1.
Step 3	**ethernet oam** **Example:**	Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-if)# ethernet oam`	
Step 4	**profile** *profile-name* **Example:** `RP/0/RP0/CPU0:router(config-if-eoam)# profile Profile_1`	Attaches the specified Ethernet OAM profile (*profile-name*), 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.

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

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure terminal`	Enters global configuration mode.
Step 2	**interface** [HundredGigE | TenGigE] *interface-path-id* **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 *rack/slot/module/port*. **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	*interface-Ethernet-OAM-command* **Example:** `RP/0/RP0/CPU0:router(config-if-eoam)# action capabilities-conflict error-disable-interface`	Configures a setting for an Ethernet OAM configuration command and overrides the setting for the profile configuration, where *interface-Ethernet-OAM-command* is one of the supported commands on the platform in interface Ethernet OAM configuration mode.
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
```

Configuration Examples for Ethernet OAM

This section provides the following configuration examples:

Configuration Examples for EOAM Interfaces

This section provides the following configuration examples:

Configuring an Ethernet OAM Profile Globally: Example

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

```
configure terminal
 ethernet oam profile Profile_1
  link-monitor
   symbol-period window 60000
   symbol-period threshold ppm low 10000000 high 60000000
   frame window 60
   frame threshold ppm low 10000000 high 60000000
   frame-period window 60000
   frame-period threshold ppm low 100 high 12000000
   frame-seconds window 900000
   frame-seconds threshold low 3 high 900
   exit
  mib-retrieval
  connection timeout 30
  require-remote mode active
  require-remote mib-retrieval
  action dying-gasp error-disable-interface
  action critical-event error-disable-interface
  action discovery-timeout error-disable-interface
  action session-down error-disable-interface
  action capabilities-conflict error-disable-interface
  action wiring-conflict error-disable-interface
  action remote-loopback error-disable-interface
  commit
```

Configuring Ethernet OAM Features on an Individual Interface: Example

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

```
configure terminal
 interface TenGigE 0/1/0/0
  ethernet oam
   link-monitor
    symbol-period window 60000
    symbol-period threshold ppm low 10000000 high 60000000
    frame window 60
    frame threshold ppm low 10000000 high 60000000
    frame-period window 60000
    frame-period threshold ppm low 100 high 12000000
    frame-seconds window 900000
    frame-seconds threshold low 3 high 900
    exit
   mib-retrieval
   connection timeout 30
   require-remote mode active
   require-remote mib-retrieval
   action link-fault  error-disable-interface
   action dying-gasp error-disable-interface
   action critical-event error-disable-interface
   action discovery-timeout error-disable-interface
   action session-down error-disable-interface
   action capabilities-conflict error-disable-interface
   action wiring-conflict error-disable-interface
   action remote-loopback 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 terminal
 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
  action remote-loopback disable
  action uni-directional link-fault error-disable-interface
  commit

configure terminal
 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
    action remote-loopback log
    action uni-directional link-fault log
    uni-directional link-fault detection
    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 terminal
 snmp-server traps ethernet oam events
```

Ethernet CFM

Table 1: Feature History Table

Feature Name	Release	Description
Cisco NC57 Native Mode: CFM	Release 7.3.1	This feature is now supported on routers that have the Cisco NC57 line cards installed and operate in the native mode. To enable the native mode, use the **hw-module profile npu native-mode-enable** command in the configuration mode. Ensure that you reload the router after configuring the native mode.
Cisco NC57 Compatibility Mode: CFM	Release 7.4.1	This feature is now supported on routers that have the Cisco NC57 line cards installed and operate in the compatibility mode.

Feature Name	Release	Description
Support for Link Loss Forwarding on Cisco NCS 5500 Series Routers	Release 7.5.1	This feature, now available on Cisco NCS 5500 Series Routers, enables high availability between two bridged interfaces by disabling both interfaces if any one of them fails. This functionality allows a fault detected on one side of a CFM-protected network to propagate to the other side, enabling the device to re-route around the failure at that end. In earlier releases, a failure on one bridged interface did not disable the other interface, and connected devices remained unaware of the link loss.

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.

Note Enable a maximum of 32 VLAN ranges per NPU. Else, when you reload the device, all CFM sessions over the 802.1Q VLAN interface might go down. Also, the corresponding bundle interface might go down. If more than 32 VLAN ranges exist on an NPU, remove the additional VLAN ranges and reload the device to address the issue. This is not applicable for NCS 5700 line cards.

Note Up MEP with Cisco NC57 line cards installed and operate in the native and compatibility modes as a part of Layer 2 service. When you have have NC57 line card (compatibility mode) interface as core facing (ingress) and NC57 line card as the AC (egress) interface, the up mep CFM session does not come up.

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, ETH-BNM, ETH-CSF—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.

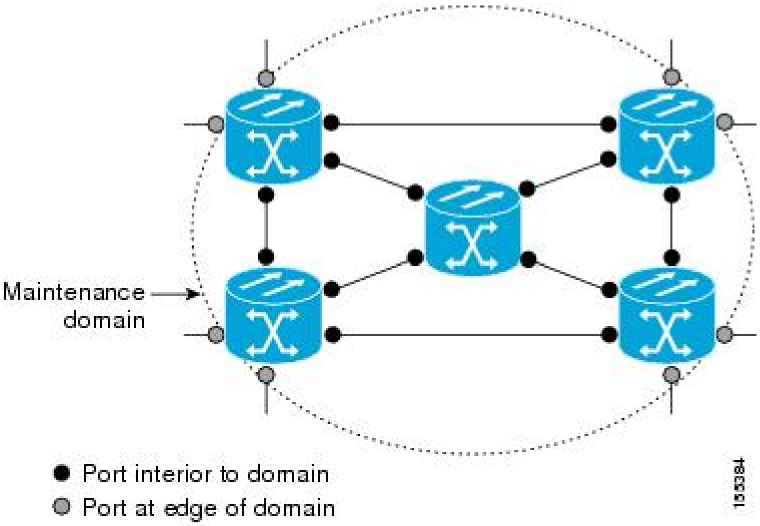
- ETH-AIS—The reception of ETH-LCK messages is also supported.

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

Maintenance Domains

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:

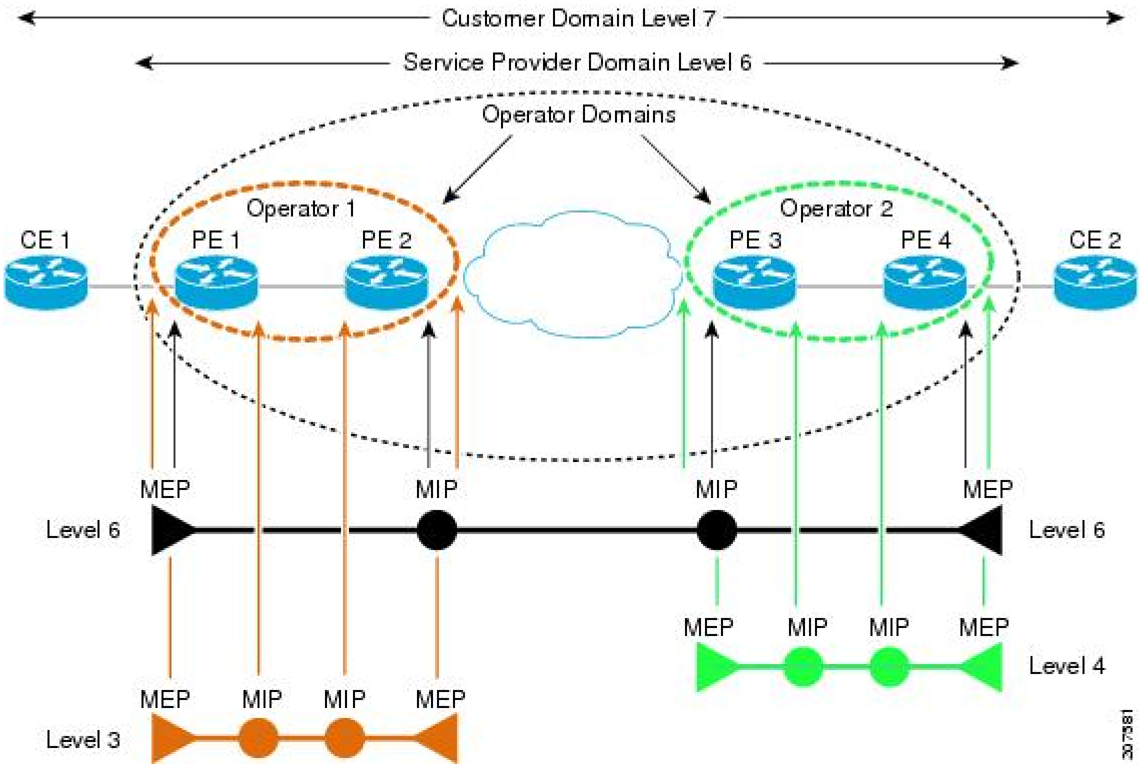
- The customer can use CFM between their CE devices, to verify and manage connectivity across the whole network.
- The service provider can use CFM between their PE devices, to verify and manage the services they are providing.
- Each operator can use CFM within their operator network, to verify and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

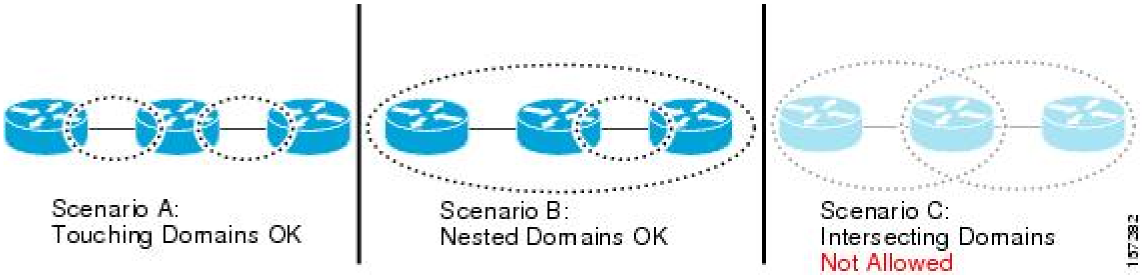
Figure 3: Different CFM Maintenance Domains Across a Network



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.

Figure 4: Supported CFM Maintenance Domain Structure



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 two types of MP:

- 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.
- Maintenance Intermediate Points (MIPs)—Created in the middle of the domain. Unlike MEPS, MIPs do allow CFM frames at their own level to be forwarded.

MIP Creation

Unlike MEPs, MIPs are not explicitly configured on each interface. MIPs are created automatically according to the algorithm specified in the CFM 802.1ag standard. The algorithm, in brief, operates as follows for each interface:

- The bridge-domain or cross-connect for the interface is found, and all services associated with that bridge-domain or cross-connect are considered for MIP auto-creation.

- The level of the highest-level MEP on the interface is found. From among the services considered above, the service in the domain with the lowest level that is higher than the highest MEP level is selected. If there are no MEPs on the interface, the service in the domain with the lowest level is selected.
- The MIP auto-creation configuration (**mip auto-create** command) for the selected service is examined to determine whether a MIP should be created.

Note Configuring a MIP auto-creation policy for a service does not guarantee that a MIP will automatically be created for that service. The policy is only considered if that service is selected by the algorithm first.

MEP and CFM Processing Overview

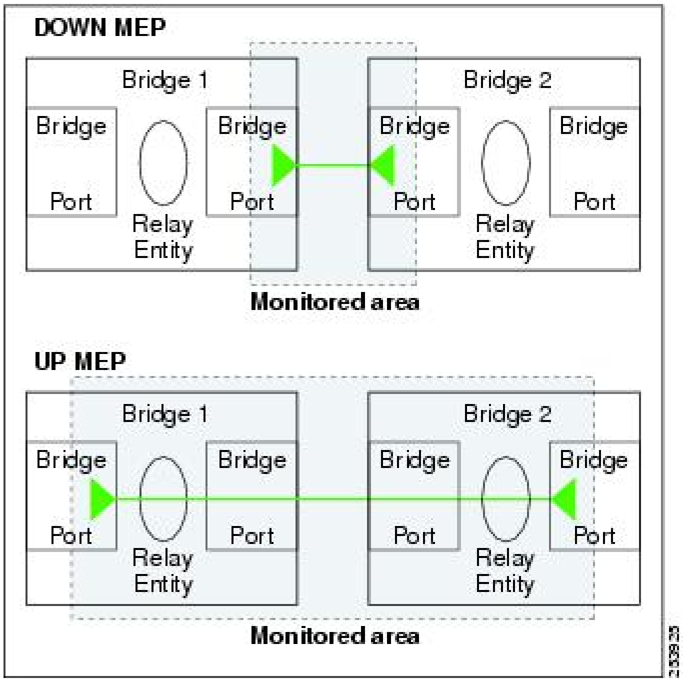
The boundary of a domain is an interface, rather than a bridge or host. Therefore, MEPs can be sub-divided into two categories:

- 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.

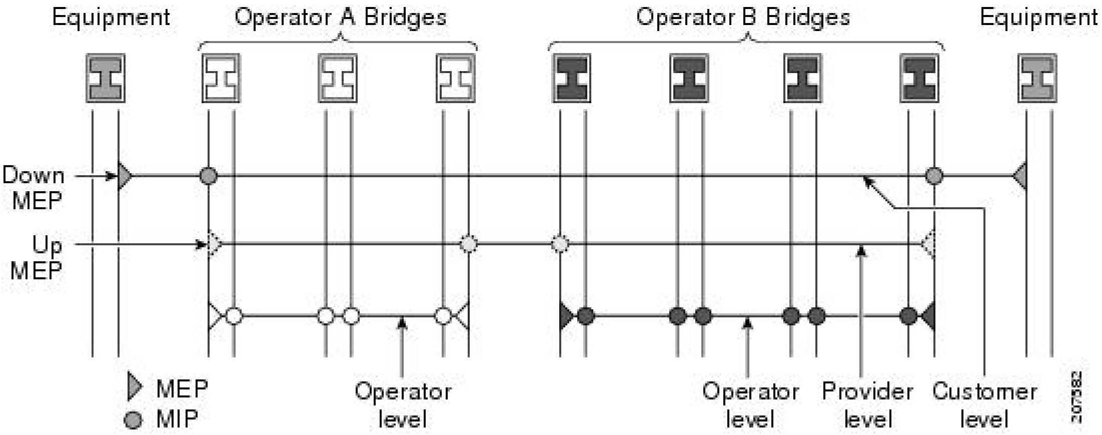
Note The terms *Down MEP* and *Up MEP* are defined in the IEEE 802.1ag and ITU-T Y.1731 standards, and refer to the direction that CFM frames are sent from the MEP. The terms should not be confused with the operational status of the MEP.

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

Figure 5: Monitored Areas for Down and Up MEPs



This figure shows maintenance points at different levels. Because domains are allowed to nest but not intersect (see Supported CFM Maintenance Domain Structure), a MEP at a low level always corresponds with a MEP or MIP at a higher level. In addition, only a single MIP is allowed on any interface—this is generally created in the lowest domain that exists at the interface and that does not have a MEP.



MIPs and 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.

MIPs also continue to receive CFM frames at their level if the interface is STP blocked, and can respond to any received frames. However, MIPs do not allow CFM frames at the level of the MIP to be forwarded if the interface is blocked.

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.

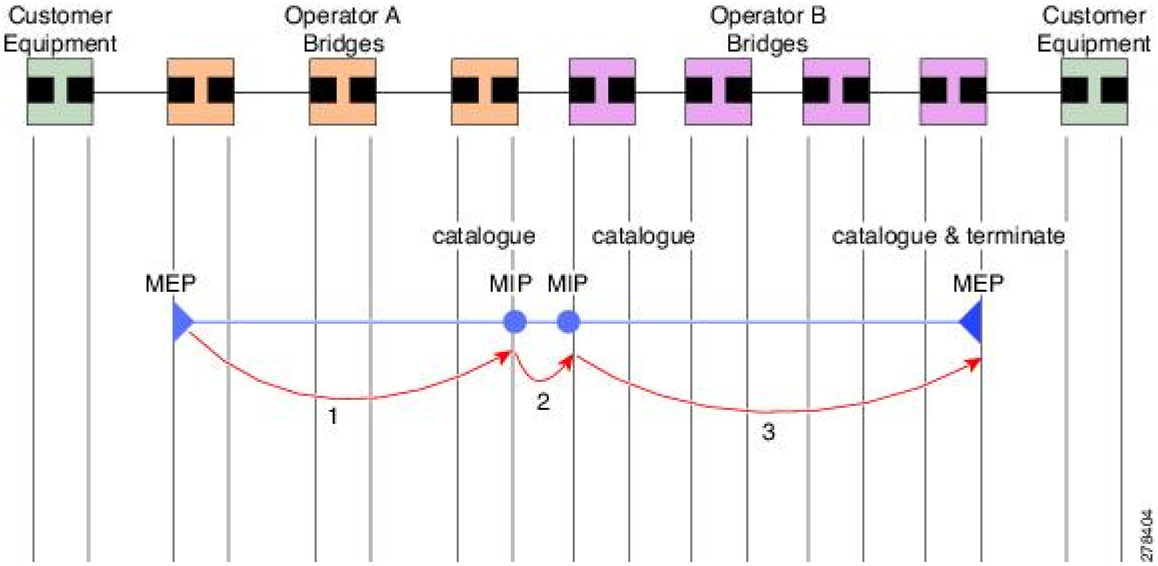
This section describes the following CFM messages:

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.

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).

Figure 6: Continuity Check Message Flow



All the MEPs in a service must transmit CCMs at the same interval. The interval at which CCMs are being transmitted is called CCM interval. IEEE 802.1ag defines 7 possible intervals that can be used:

- 3.3ms
- 10ms
- 100ms
- 1s
- 10s
- 1 minute
- 10 minutes

A MEP detects a loss of connectivity with one of its peer MEPs when a certain number of CCMs have been 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.

CFM is supported only on interfaces which have Layer 2 transport feature enabled.

Maintenance Association Identifier (MAID)

Table 2: Feature History Table

Feature Name	Release	Description
48 byte string-based MAID support for Offloaded Endpoints	Release 7.5.1	This feature is supported on Cisco Network Convergence System 5700 Series routers and routers with the Cisco NC57 line cards operating in native mode. This feature extends MAID functionality to support the flexible format for hardware offloaded MEPs. This removes the restrictions on the type of MAID supported for sessions with less than 1 minute time intervals. To enable the feature in native mode, run the Ethernet CFM Show Command for flexible packet format: Examples command in the System Admin Config mode, and reload the router.

Continuity Check Messages (CCM) are essential for detecting various defects in network services. They carry critical information that helps in the identification and maintenance of the service. This is a breakdown of the information contained in CCM messages:

- **Maintenance Domain Identifier (MDID):** A configured identifier unique to the domain of the transmitting Maintenance End Point (MEP). It is crucial for the identification of the maintenance domain.

- **Short MA Name (SMAN):** A configured identifier specific to the service of the transmitting MEP. It is used to identify the service within the maintenance domain.
- **Maintenance Association Identifier (MAID):** A combination of MDID and SMAN. Together, these identifiers form the MAID, which is a composite identifier that must be uniformly configured across all MEPs within the same service.

Note MDID **only** supports **null** value and SMAN supports ITU Carrier Code (ICC) or a numerical. No other values are supported.

Supported MAID Formats for Offloaded MEPs (applicable for NCS 5700 line cards only)

- 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

These are some examples:

- Configuring domain ID null: **ethernet cfm domain SMB level 3 id null**
- Configuring SMAN: **ethernet cfm domain SMB level 3 id null service 901234AB xconnect group 99999 p2p 99999 id number 1**

This table summarizes the supported values and parameters for MDID and SMAN. This table only details the MAID restriction on the hardware offload feature. There is no MAID restriction for software offload or non-offloaded MEPs.

For Cisco NCS 5500 series routers, "id null" has to be explicitly configured for the domain ID, for hardware offloaded sessions.

Format	MDID	SMAN	Support	Comment
	No	2 byte integer	Yes	Up to 2000 entries

Format	MDID	SMAN	Support	Comment
	No	13 bytes ICCCode (6 bytes) and UMC (7 bytes)	Yes	Up to 15 unique ICC Up to 4K UMC values
48 bytes string based	1-48 bytes of MDID and SMAN		No	Most commonly used

Guidelines and Restrictions for MAID

- Configure each MEP within the service with a distinct MEP ID, which is a unique numeric identifier.
- Configure MEP CrossCheck for all MEPs with intervals of less than 10s, as Dynamic Remote MEPs are not supported for these.
- In a Remote Defect Indication (RDI), each MEP includes sequence number 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. Sequence numbering is not supported for MEPs with CCM intervals of less than 10s.
- CCM Tx/Rx statistics counters are not supported for MEPs with less than 10s intervals.
- Sender TLV and Cisco Proprietary TLVs are not supported for MEPs with less than 10s intervals.
- Starting from Cisco IOS XR SoftwareRelease 7.5.1, MAID supports the flexible packet format of MEG IDs on hardware offloaded MEPs for the following Cisco NC57 line cards:
    - NC57-24DD
    - NCS-57C3-MODS-SYS

  This feature is supported only on Cisco NC57 line cards installed and operate in native mode. It removes the restrictions on the type of MAID that are supported for sessions with less than 1 minute time intervals. This helps in interoperating with the devices that already support the flexible format configuration.

  Examples:

    - Configuring domain ID: **ethernet cfm domain SMB level 3 id string** or

      ethernet cfm domain SMB level 3

    - Configuring SMAN: **ethernet cfm domain SMB level 3 id null service 901234AB xconnect group 99999 p2p 99999 id string** or

      ethernet cfm domain SMB level 3 id null service 901234AB xconnect group 99999 p2p 99999

- The status of the interface where the MEP is operating (for example, up - when the interface is up, or down - when the interface is down) should not be confused with the direction of any MEPs on the interface (Up MEPs/Down MEPs).

Defect Identification using CCM Analysis

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 configured to send CCMs.
- Level mismatch: A MEP receives a CCM carrying a lower maintenance level than the MEP's own configured level.
- Loop: A CCM is received with a source MAC address that matches the MAC address of the MEP's operating interface, indicating a loop.
- Configuration error: A received CCM contains a MEP ID that duplicates the MEP ID of the receiving MEP, signaling a configuration issue.
- Cross-connect error: A CCM with a non-matching MAID is received, often pointing to a VLAN misconfiguration that causes service leakage.
- 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. This does not trigger the local MEP to send out CCMs with a remote defect indication.

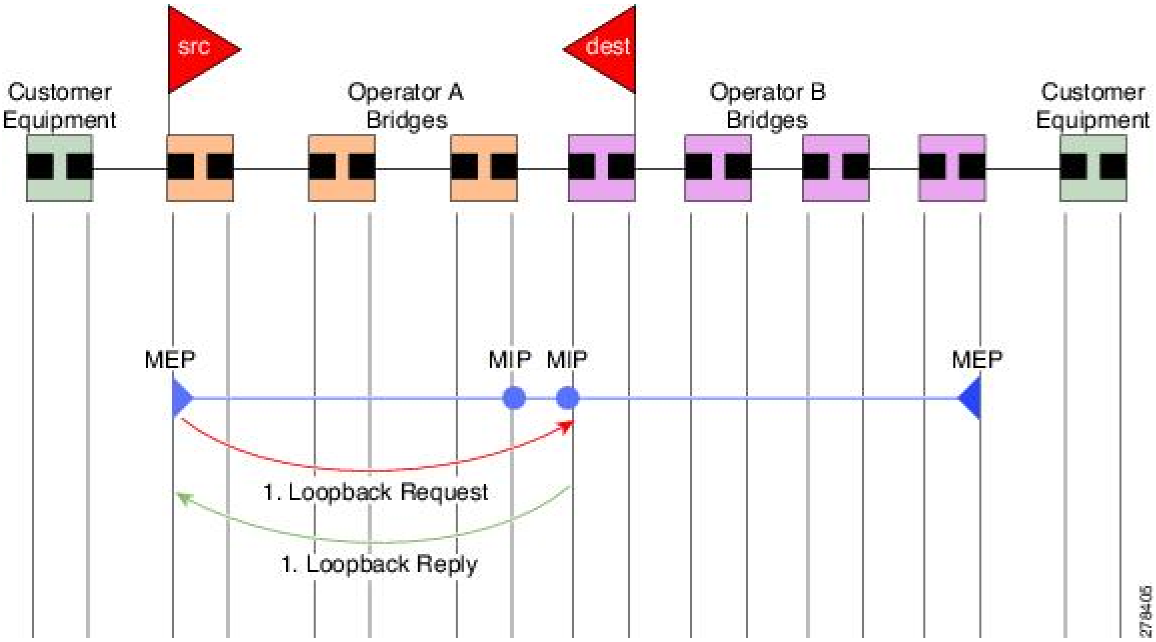
By monitoring the sequence numbers in CCMs from peer MEPs, out-of-sequence CCMs can be identified, although these are not classified as CCM defects.

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. 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 MIP.

Figure 7: Loopback Messages



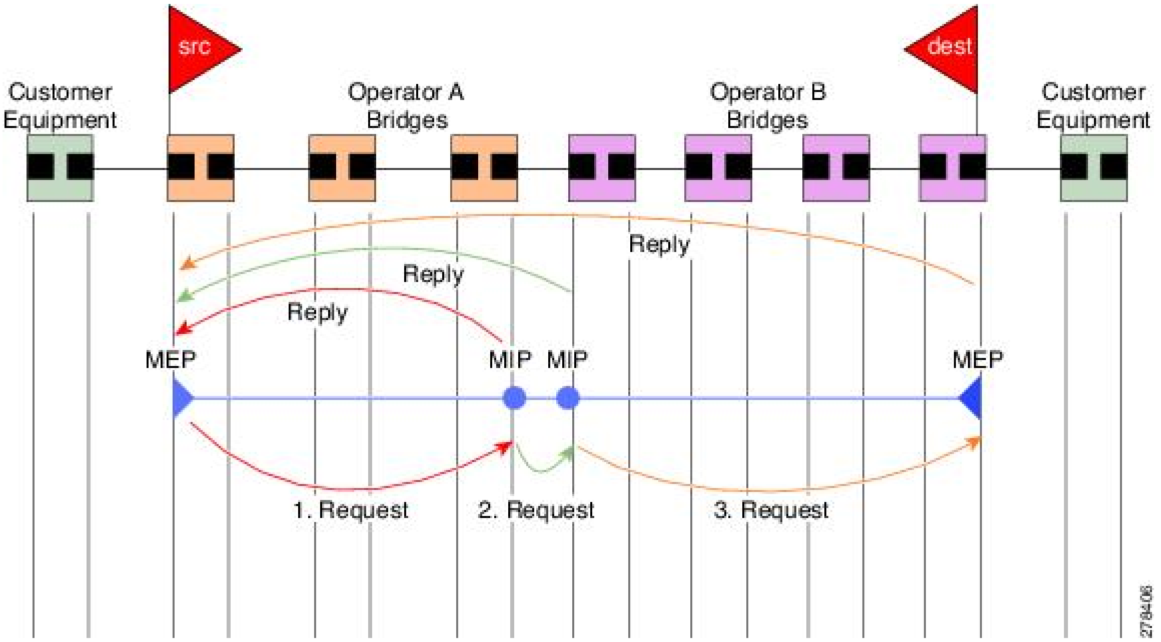
Loopback messages can be padded with user-specified data. This allows data corruption to be detected in the network. They also carry a sequence number which allows for out-of-order frames to be detected.

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. At the request of the operator, a local MEP sends an LTM. Each hop where there is a maintenance point sends an 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 MIPs.

Figure 8: 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.

Flexible VLAN Tagging for CFM

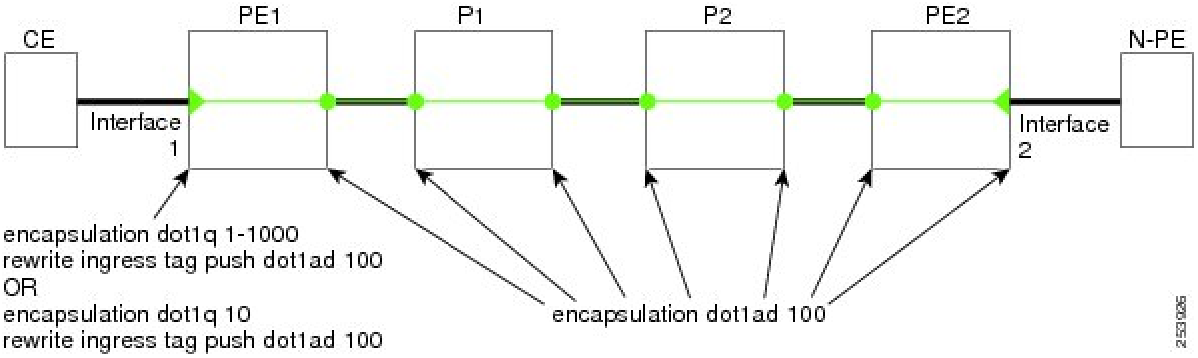
The Flexible VLAN Tagging for CFM feature ensures that CFM packets are sent with the right VLAN tags so that they are appropriately handled as a CFM packet by the remote device. When packets are received by an edge router, they are treated as either CFM packets or data packets, depending on the number of tags in the header. The system differentiates between CFM packets and data packets based on the number of tags in the packet, and forwards the packets to the appropriate paths based on the number of tags in the packet.

CFM frames are normally sent with the same VLAN tags as the corresponding customer data traffic on the interface, as defined by the configured encapsulation and tag rewrite operations. Likewise, received frames are treated as CFM frames if they have the correct number of tags as defined by the configured encapsulation and tag rewrite configuration, and are treated as data frames (that is, they are forwarded transparently) if they have more than this number of tags.

In most cases, this behavior is as desired, since the CFM frames are then treated in exactly the same way as the data traffic flowing through the same service. However, in a scenario where multiple customer VLANs are multiplexed over a single multipoint provider service (for example, N:1 bundling), a different behavior might be desirable.

This figure shows an example of a network with multiple VLANS using CFM.

Figure 9: Service Provider Network With Multiple VLANs and CFM


CE
PE1
P1
P2
PE2
N-PE
Interface 1
Interface 2
encapsulation dot1q 1-1000
rewrite ingress tag push dot1ad 100
OR
encapsulation dot1q 10
rewrite ingress tag push dot1ad 100
encapsulation dot1ad 100
253026

This figure shows a provider's access network, where the S-VLAN tag is used as the service delimiter. PE1 faces the customer, and PE2 is at the edge of the access network facing the core. N:1 bundling is used, so the interface encapsulation matches a range of C-VLAN tags. This could potentially be the full range, resulting in all:1 bundling. There is also a use case where only a single C-VLAN is matched, but the S-VLAN is nevertheless used as the service delimiter—this is more in keeping with the IEEE model, but limits the provider to 4094 services.

CFM is used in this network with a MEP at each end of the access network, and MIPs on the boxes within the network (if it is native Ethernet). In the normal case, CFM frames are sent by the up MEP on PE1 with two VLAN tags, matching the customer data traffic. This means that at the core interfaces and at the MEP on PE2, the CFM frames are forwarded as if they were customer data traffic, since these interfaces match only on the S-VLAN tag. So, the CFM frames sent by the MEP on PE1 are not seen by any of the other MPs.

Flexible VLAN tagging changes the encapsulation for CFM frames that are sent and received at Up MEPs. Flexible VLAN tagging allows the frames to be sent from the MEP on PE1 with just the S-VLAN tag that represents the provider service. If this is done, the core interfaces will treat the frames as CFM frames and they will be seen by the MIPs and by the MEP on PE2. Likewise, the MEP on PE1 should handle received frames with only one tag, as this is what it will receive from the MEP on PE2.

To ensure that CFM packets from Up MEPs are routed to the appropriate paths successfully, tags may be set to a specific number in a domain service, using the **tags** command. Currently, tags can only be set to one (1).

Configuring Ethernet CFM

To configure Ethernet CFM, perform the following tasks:

Note CFM is not supported for the following:

- L3 Interfaces and Sub-Interfaces
- Bundle Member Ports
- Bridge Domain
- CFM over BGP-VPLS is supported only for NCS 5700 line cards.

Configuring a CFM Maintenance Domain

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

SUMMARY STEPS

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

DETAILED STEPS

	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	**traceroute cache hold-time** *minutes* **size** *entries* **Example:** `RP/0/RP0/CPU0:router(config-cfm)# traceroute cache hold-time 1 size 3000`	(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 4	**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 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.

Step 5	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cfm-dmn)# commit`
		Saves configuration changes.
		• When you use the **end** command, the system prompts you to commit changes:
		`Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:`

	Command or Action	Purpose
		• 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

You can configure up to 32000 CFM services for a maintenance domain. 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* **m2mp | p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [[**number** *number*]
5. **end** or **commit**

DETAILED STEPS

	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:**	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

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1`	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* **m2mp** | **p2p** *xconnect-name*}[**id** [**icc-based** *icc-string umc-string*] | [[**number** *number*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service ABC xconnect group X1 p2p ADB`	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 bridge domain where MIPs and up MEPs will be created. 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:
		`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.

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

	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** *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 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** *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 ABC xconnect group X1 p2p ADB`	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 bridge domain or xconnect where MIPs and up MEPs will be created. The **id** sets the short MA name.
Step 5	**continuity-check interval** *time* [**loss-threshold** *threshold*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check interval 100m loss-threshold 10`	(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:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check archive hold-time 100`	(Optional) Configures how long information about peer MEPs is stored after they have timed out.
Step 7	**continuity-check loss auto-traceroute** **Example:**	(Optional) Configures automatic triggering of a traceroute when a MEP is declared down.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# continuity-check loss auto-traceroute`	

Step 8	**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:
		`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.

Configuring Automatic MIP Creation for a CFM Service

For more information about the algorithm for creating MIPs, see the **MIP Creation** section.

To configure automatic MIP creation 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. **mip auto-create** {**all** | **lower-mep-only**} {**ccm-learning**}
6. **end** or **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`RP/0/RP0/CPU0:router# configure`	
Step 2	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router# ethernet cfm`	Enters the Ethernet Connectivity Fault Management (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 and enters the CFM domain configuration mode. The level must be specified. The only supported option is **id [null]** for less than 1min interval MEPS. 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 ABC xconnect group X1 p2p ADB`	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 bridge domain where MIPs and up MEPs will be created. The **id** sets the short MA name.

Step 5	**mip auto-create** {**all** | **lower-mep-only**} {**ccm-learning**}
	Example:
	`RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mip auto-create all ccm-learning`
		(Optional) Enables the automatic creation of MIPs in a bridge domain. **ccm-learning** option enables CCM learning for MIPs created in this service. This must be used only in services with a relatively long CCM interval of at least 100 ms. CCM learning at MIPs is disabled by default.

Step 6	**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:
		`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.

	Command or Action	Purpose
		• 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* {**bridge group** *bridge-domain-group* **bridge-domain** *bridge-domain-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* [**mac-address** *mac-address*]
7. **end** or **commit**

DETAILED STEPS

	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** *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 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* {**bridge group** *bridge-domain-group* **bridge-domain** *bridge-domain-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*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1`	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 bridge domain or xconnect where MIPs and up MEPs will be created. The **id** sets the short MA name.
Step 5	**mep crosscheck** **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mep crosscheck mep-id 10`	Enters CFM MEP crosscheck configuration mode.
Step 6	**mep-id** *mep-id-number* [**mac-address** *mac-address*] **Example:** `RP/0/RP0/CPU0:router(config-cfm-xcheck)# mep-id 10`	Enables cross-check on a MEP. **Note** • 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:
	`RP/0/RP0/CPU0:router(config-cfm-xcheck)# commit`
		Saves configuration changes.
		• When you use 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.

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* {**bridge group** *bridge-domain-group* **bridge-domain** *bridge-domain-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

	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** *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 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** *service-name* {**bridge group** *bridge-domain-group* **bridge-domain** *bridge-domain-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*]] **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1`	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 bridge domain or xconnect where MIPs and up MEPs will be created. The **id** sets the short MA name.
Step 5	**maximum-meps** *number* **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:
		`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.

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

1. **configure**
2. **interface** {**HundredGigE** | **TenGigE**} *interface-path-id*
3. **interface** {**HundredGigE** | **TenGigE** | **Bundle-Ether**} *interface-path-id***.***subinterface*
4. **vrf vrf-name**
5. **interface** {**HundredGigE** | **TenGigE**} *interface-path-id*
6. **ethernet cfm**
7. **mep domain** *domain-name* **service** *service-name* **mep-id** *id-number*
8. **cos** *cos*
9. **end** or **commit**

DETAILED STEPS

	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.
Step 3	**interface** {**HundredGigE** | **TenGigE** | **Bundle-Ether**} *interface-path-id***.***subinterface* **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 subinterface path ID. Naming convention is *interface-path-id***.***subinterface*. The period in front of the subinterface value is required as part of the notation.
Step 4	**vrf vrf-name** **Example:** `RP/0/RP0/CPU0:router(config-if)# vrf vrf_A`	Configures a VRF instance and enters VRF configuration mode.
Step 5	**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.
Step 6	**ethernet cfm** **Example:** `RP/0/RP0/CPU0:router(config-if)# ethernet cfm`	Enters interface Ethernet CFM configuration mode.
Step 7	**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.

	Command or Action	Purpose
Step 8	**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 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 9	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-if-cfm-mep)# commit`
		Saves configuration changes.
		• When you use 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.

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* **bridge group** *name* **bridge-domain** *name*
5. **service** *name* **xconnect group** *xconnect-group-name* **p2p** *xconnect-name*
6. **ais transmission** [**interval** {**1s**|**1m**}][**cos** *cos*]
7. **log ais**
8. **end** or **commit**

DETAILED STEPS

	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* **bridge group** *name* **bridge-domain** *name* **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2`	Specifies the service, bridge group, and bridge domain.
Step 5	**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 6	**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 7	**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.

	Command or Action	Purpose

Step 8	**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.

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

	Command or Action	Purpose
Step 1	**configure** **Example:** `RP/0/RP0/CPU0:router# configure`	Enters global configuration mode.
Step 2	**interface gigabitethernet** *interface-path-id* **Example:** `RP/0/RP0/CPU0:router# interface TenGigE 0/0/0/2`	Enters interface configuration mode.

	Command or Action	Purpose
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** *cos* **Example:** `RP/0/RP0/CPU0:router(config-if-cfm)# ais transmission up interval 1m cos 7`	Configures Alarm Indication Signal (AIS) transmission on a Connectivity Fault Management (CFM) interface.

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.

Configuring Flexible VLAN Tagging for CFM

Use this procedure to set the number of tags in CFM packets in a CFM domain service.

SUMMARY STEPS

1. **configure**
2. **ethernet cfm**
3. **domain** *name* **level** *level*
4. **service** *name* **bridge group** *name* **bridge-domain** *name*
5. **tags** *number*
6. **end** or **commit**

DETAILED STEPS

	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* **bridge group** *name* **bridge-domain** *name* **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn)# service S2 bridge group BG1 bridge-domain BD2`	Specifies the service, bridge group, and bridge domain.
Step 5	**tags** *number* **Example:** `RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# tags 1`	Specifies the number of tags in CFM packets. Currently, the only valid value is 1.

Step 6	**end** or **commit**
	Example:
	`RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# 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.

	Command or Action	Purpose
		• 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.

Troubleshooting Tips

To troubleshoot problems within the CFM network, perform these steps:

SUMMARY STEPS

1. To verify connectivity to a problematic MEP, use the **ping ethernet cfm** command as shown in this example:
2. If the results of the **ping ethernet cfm** command show a problem with connectivity to the peer MEP, use the **traceroute ethernet cfm** command to help further isolate the location of the problem as shown in the following example:

DETAILED STEPS

Step 1 To verify connectivity to a problematic MEP, use the **ping ethernet cfm** command as shown in this example:

```
RP/0/RP0/CPU0:router# ping ethernet cfm domain D1 service S1 mep-id 16 source
interface TenGigE 0/0/0/1

Type escape sequence to abort.
Sending 5 CFM Loopbacks, timeout is 2 seconds -
Domain foo (level 2), Service foo
Source: MEP ID 1, interface TenGigE0/0/0/1
Target: 0001.0002.0003 (MEP ID 16):
  Running (5s) ...
Success rate is 60.0 percent (3/5), round-trip min/avg/max = 1251/1349/1402 ms
Out-of-sequence: 0.0 percent (0/3)
```

```
Bad data: 0.0 percent (0/3)
Received packet rate: 1.4 pps
```

Step 2 If the results of the **ping ethernet cfm** command show a problem with connectivity to the peer MEP, use the **traceroute ethernet cfm** command to help further isolate the location of the problem as shown in the following example:

```
RP/0/RP0/CPU0:router# traceroute ethernet cfm domain D1 service S1 mep-id 16 source
interface TenGigE 0/0/0/2

Traceroutes in domain D1 (level 4), service S1
Source: MEP-ID 1, interface TenGigE0/0/0/2
================================================================================
Traceroute at 2009-05-18 12:09:10 to 0001.0203.0402,
TTL 64, Trans ID 2:

Hop Hostname/Last            Ingress MAC/name       Egress MAC/Name        Relay
--- ------------------------ ---------------------- ---------------------- -----
  1 ios                      0001.0203.0400 [Down]                         FDB
      0000-0001.0203.0400    TenGigE0/0/0/2
  2 abc                                             0001.0203.0401 [Ok]    FDB
      ios                                           Not present
  3 bcd                      0001.0203.0402 [Ok]                           Hit
      abc                    TenGigE0/0
Replies dropped: 0
```

If the target was a MEP, verify that the last hop shows “Hit” in the Relay field to confirm connectivity to the peer MEP.

If the Relay field contains “MPDB” for any of the hops, then the target MAC address was not found in the bridge MAC learning table at that hop, and the result is relying on CCM learning. This result can occur under normal conditions, but it can also indicate a problem. If you used the **ping ethernet cfm** command before using the **traceroute ethernet cfm** command, then the MAC address should have been learned. If “MPDB” is appearing in that case, then this indicates a problem at that point in the network.

Configuration Examples for Ethernet CFM

This section includes the following examples:

Ethernet CFM Domain Configuration: Example

This example shows how to configure a basic domain for Ethernet CFM:

```
configure
 ethernet cfm
  traceroute cache hold-time 1 size 3000
  domain Domain_One level 1 id string D1
  commit
```

Ethernet CFM Service Configuration: Example

This example shows how to create a service for an Ethernet CFM domain:

```
  service Bridge_Service bridge group BD1 bridge-domain B1
  service Cross_Connect_1 xconnect group XG1 p2p X1
  commit
```

Flexible Tagging for an Ethernet CFM Service Configuration: Example

This example shows how to set the number of tags in CFM packets from down MEPs in a CFM domain service:

```
configure
 ethernet cfm
  domain D1 level 1
   service S2 bridge group BG1 bridge-domain BD2
    tags 1
    commit
```

Continuity Check for an Ethernet CFM Service Configuration: Example

This example shows how to configure continuity-check options for an Ethernet CFM service:

```
   continuity-check archive hold-time 100
   continuity-check loss auto-traceroute
   continuity-check interval 100ms loss-threshold 10
   commit
```

MIP Creation for an Ethernet CFM Service Configuration: Example

This example shows how to enable MIP auto-creation for an Ethernet CFM service:

```
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# mip auto-create all
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# commit
```

Cross-check for an Ethernet CFM Service Configuration: Example

This example shows how to configure cross-check for MEPs in an Ethernet CFM service:

```
   mep crosscheck
    mep-id 10
    mep-id 20
    commit
```

Other Ethernet CFM Service Parameter Configuration: Example

This example shows how to configure other Ethernet CFM service options:

```
    maximum-meps 4000
    log continuity-check errors
    commit
    exit
   exit
  exit
```

MEP Configuration: Example

This example shows how to configure a MEP for Ethernet CFM on an interface:

```
 interface TenGigE 0/0/0/1
  ethernet cfm
  mep domain Dm1 service Sv1 mep-id 1
  commit
```

Ethernet CFM Show Command: Examples

These examples show how to verify the configuration of Ethernet Connectivity Fault Management (CFM):

Example 1

This example shows how to display all the maintenance points that have been created on an interface:

```
RP/0/RP0/CPU0:router# show ethernet cfm local maintenance-points

Domain/Level         Service             Interface         Type   ID   MAC
-------------------- ------------------- ----------------- ------ ---- --------
fig/5                bay                 Gi0/10/0/12       Dn MEP    2 44:55:66
fig/5                bay                 Gi0/0/1/0         MIP         55:66:77
fred/3               barney              Gi0/1/0/0         Dn MEP    5 66:77:88!
```

Example 2

This example shows how to display all the CFM configuration errors on all domains:

```
RP/0/RP0/CPU0:router# show ethernet cfm configuration-errors

Domain fig (level 5), Service bay
* MIP creation configured using bridge-domain blort, but bridge-domain blort does not exist.

 * An Up MEP is configured for this domain on interface TenGigE0/0/0/3 and an Up MEP is
also configured for domain blort, which is at the same level (5).
 * A MEP is configured on interface TenGigE0/0/0/1 for this domain/service, which has CC
interval 100ms, but the lowest interval supported on that interface is 1s
```

Example 3

This example shows how to display operational state for local maintenance end points (MEPs):

```
RP/0/RP0/CPU0:router# show ethernet cfm local meps

 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

Domain foo (level 6), Service bar
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
  100 Gi1/1/0/1 (Up)       Up     0/0   N  A       L7

Domain fred (level 5), Service barney
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
    2 Gi0/1/0/0 (Up)       Up     3/2   Y  RPC     L6
Domain foo (level 6), Service bar
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
  100 Gi1/1/0/1 (Up)       Up     0/0   N  A

Domain fred (level 5), Service barney
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
```

```
----- ----------------------- --- -------- -- ------- ---
    2 Gi0/1/0/0 (Up)       Up     3/2   Y  RPC
```

Example 4

This example shows how to display operational state of other maintenance end points (MEPs) detected by a local MEP:

```
RP/0/RP0/CPU0:router# show ethernet cfm peer meps

Flags:
 > - Ok                             I - Wrong interval
 R - Remote Defect received         V - Wrong level
 L - Loop (our MAC received)        T - Timed out
 C - Config (our ID received)       M - Missing (cross-check)
 X - Cross-connect (wrong MAID)     U - Unexpected (cross-check)

Domain fred (level 7), Service barney
Down MEP on TenGigE0/0/0/1, MEP-ID 2
================================================================================
St    ID MAC address    Port    Up/Downtime   CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ----------- --------- ------ ----- -----
 >     1 0011.2233.4455 Up      00:00:01         1234      0     0     0
R>     4 4455.6677.8899 Up      1d 03:04         3456      0   234     0
L      2 1122.3344.5566 Up      3w 1d 6h         3254      0     0  3254
C      2 7788.9900.1122 Test    00:13            2345      6    20  2345
X      3 2233.4455.6677 Up      00:23              30      0     0    30
I      3 3344.5566.7788 Down    00:34           12345      0   300  1234
V      3 8899.0011.2233 Blocked 00:35              45      0     0    45
 T     5 5566.7788.9900         00:56              20      0     0     0
M      6                                            0      0     0     0
U>     7 6677.8899.0011 Up      00:02             456      0     0     0

Domain fred (level 7), Service fig
Down MEP on TenGigE0/0/0/12, MEP-ID 3
================================================================================
St    ID MAC address    Port    Up/Downtime  CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ----------- -------- ------ ----- -----
 >     1 9900.1122.3344 Up      03:45           4321      0     0     0
```

Example 5

This example shows how to display operational state of other maintenance end points (MEPs) detected by a local MEP with details:

```
RP/0/RP0/CPU0:router#  show ethernet cfm peer meps detail
Domain dom3 (level 5), Service ser3
Down MEP on TenGigE0/0/0/1 MEP-ID 1
================================================================================
Peer MEP-ID 10, MAC 0001.0203.0403
   CFM state: Wrong level, for 00:01:34
   Port state: Up
   CCM defects detected:    V - Wrong Level
   CCMs received: 5
     Out-of-sequence:             0
     Remote Defect received:      5
     Wrong Level:                 0
     Cross-connect (wrong MAID):  0
     Wrong Interval:              5
     Loop (our MAC received):     0
     Config (our ID received):    0
```

```
Last CCM received 00:00:06 ago:
     Level: 4, Version: 0, Interval: 1min
     Sequence number: 5, MEP-ID: 10
     MAID: String: dom3, String: ser3
     Port status: Up, Interface status: Up


Domain dom4 (level 2), Service ser4
Down MEP on TenGigE0/0/0/2 MEP-ID 1
================================================================================
Peer MEP-ID 20, MAC 0001.0203.0402
   CFM state: Ok, for 00:00:04
   Port state: Up
   CCMs received: 7
     Out-of-sequence:             1
     Remote Defect received:      0
     Wrong Level:                 0
     Cross-connect (wrong MAID):  0
     Wrong Interval:              0
     Loop (our MAC received):     0
  Config (our ID received):    0
Last CCM received 00:00:04 ago:
     Level: 2, Version: 0, Interval: 10s
     Sequence number: 1, MEP-ID: 20
     MAID: String: dom4, String: ser4
     Chassis ID: Local: ios; Management address: 'Not specified'
     Port status: Up, Interface status: Up

Peer MEP-ID 21, MAC 0001.0203.0403
   CFM state: Ok, for 00:00:05
   Port state: Up
   CCMs received: 6
     Out-of-sequence:             0
     Remote Defect received:      0
     Wrong Level:                 0
     Cross-connect (wrong MAID):  0
     Wrong Interval:              0
     Loop (our MAC received):     0
     Config (our ID received):    0
Last CCM received 00:00:05 ago:
     Level: 2, Version: 0, Interval: 10s
     Sequence number: 1, MEP-ID: 21
     MAID: String: dom4, String: ser4
     Port status: Up, Interface status: Up


Peer MEP-ID 601, MAC 0001.0203.0402
   CFM state: Timed Out (Standby), for 00:15:14, RDI received
   Port state: Down
   CCM defects detected:    Defects below ignored on local standby MEP
                            I - Wrong Interval
                            R - Remote Defect received
                            T - Timed Out
                            P - Peer port down
   CCMs received: 2
     Out-of-sequence:             0
     Remote Defect received:      2
     Wrong Level:                 0

     Wrong Interval:              2
     Loop (our MAC received):     0
     Config (our ID received):    0
   Last CCM received 00:15:49 ago:
     Level: 2, Version: 0, Interval: 10s
```

```
Sequence number: 1, MEP-ID: 600
MAID: DNS-like: dom5, String: ser5
Chassis ID: Local: ios; Management address: 'Not specified'
Port status: Up, Interface status: Down
```

Ethernet CFM Show Command for flexible packet format: Examples

The flexible packet format supports the following two types of formats:

- MDID String format
- MDID Invalid format

Note To enable the feature in native mode, use the **hw-module profile oam 48byte-cfm-maid-enable** command in the System Admin Config mode. Ensure that you reload the router after configuring the native mode.

```
Router(config)#hw-module profile oam ?
  48byte-cfm-maid-enable  Enable 48byte cfm maid feature
  sat-enable              enable SAT feature
Router(config)#hw-module profile oam 48byte-cfm-maid-enable
In order to make the oam profile take effect, the router must be manually reloaded.
Router(config)#commit

Router(config)#hw-module profile npu native-mode-enable
Tue Nov 16 06:48:34.027 UTC
In order to activate this new npu profile, you must manually reload the chassis
Router(config)#commit
```

MDID String format: Example

Configuration

```
Router(config)#ethernet cfm
Router(config-cfm)#domain test level 3 id string test_domain
Router(config-cfm-dmn)#service test down-meps id string test_service
Router(config-cfm-dmn-svc)#mep crosscheck mep-id 4
Router(config-cfm-dmn-svc)#log continuity-check mep changes
Router(config-cfm-dmn-svc)#continuity-check interval 10ms
Router(config-cfm-dmn-svc)#commit
Router(config-cfm-dmn-svc)#root
Router(config)#interface TenGigE0/0/0/0.1 l2tr


Router(config-subif)#encapsulation dot1q 1


Router(config-subif)#ethernet cfm
Router(config-if-cfm)#mep domain test service test mep-id 3
Router(config-if-cfm-mep)#commit
```

Verification

```
Router#show ethernet cfm peer meps

Tue Nov 16 06:46:13.859 UTC
Flags:
```

```
> - Ok                          I - Wrong interval
R - Remote Defect received      V - Wrong level
L - Loop (our MAC received)     T - Timed out
C - Config (our ID received)    M - Missing (cross-check)
X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
* - Multiple errors received    S - Standby

Domain test (level 3), Service test
Down MEP on TenGigE0/0/0/0.1 MEP-ID 3
================================================================================
St    ID MAC Address    Port    Up/Downtime   CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ----------- --------- ------ ----- -----
>     4 d46a.355c.b814 Up      00:02:30            0      0     0     0

Router#show ethernet cfm peer meps detail
Tue Nov 16 06:46:29.169 UTC
Domain test (level 3), Service test
Down MEP on TenGigE0/0/0/0.1 MEP-ID 3
================================================================================
Peer MEP-ID 4, MAC d46a.355c.b814
   CFM state: Ok, for 00:02:46
   Received CCM handling offloaded to hardware
   Port state: Up
   CCMs received: 0
     Out-of-sequence:             0
     Remote Defect received:      0
     Wrong level:                 0
     Cross-connect (wrong MAID):  0
     Wrong interval:              0
     Loop (our MAC received):     0
     Config (our ID received):    0
   Last CCM received:
     Level: 3, Version: 0, Interval: 10ms
     Sequence number: 0, MEP-ID: 4
     MAID: String: test_domain, String: test_service
     Port status: Up, Interface status: Up


Router#show ethernet cfm local meps verbose
Tue Nov 16 06:46:41.783 UTC
Domain test (level 3), Service test
Down MEP on TenGigE0/0/0/0.1 MEP-ID 3
================================================================================
  Interface state: Up      MAC address: b0c5.3cff.c080
  Peer MEPs: 1 up, 0 with errors, 0 timed out (archived)
  Cross-check errors: 0 missing, 0 unexpected

  CCM generation enabled:  Yes, 10ms (Remote Defect detected: No)
                           CCM processing offloaded to hardware
  AIS generation enabled:  No
  Sending AIS:             No
  Receiving AIS:           No
  Sending CSF:             No
  Receiving CSF:           No

  No packets sent/received


Router#
Router#show run interface tenGigE 0/0/0/0.1
Tue Nov 16 06:47:09.035 UTC
interface TenGigE0/0/0/0.1 l2transport
encapsulation dot1q 1
ethernet cfm
  mep domain test service test mep-id 3
  !
!
!

Router#show run ethernet cfm
Tue Nov 16 06:47:23.800 UTC
ethernet cfm
domain test level 3 id string test_domain
  service test down-meps id string test_service
   continuity-check interval 10ms
   mep crosscheck
    mep-id 4
   !
   log continuity-check mep changes
  !
!
!
-----------------------------------------------------------------------
```

MDID Invalid format: Example

Configuration

```
Router#show run ethernet cfm

Tue Nov 16 06:57:14.099 UTC

ethernet cfm
domain test level 3
  service test down-meps
   continuity-check interval 10ms
   mep crosscheck
    mep-id 4
   !
   log continuity-check mep changes
  !
!
!
```

Verification

```
Router#show ethernet  cfm peer meps
Tue Nov 16 06:57:19.027 UTC
Flags:
> - Ok                          I - Wrong interval
R - Remote Defect received      V - Wrong level
L - Loop (our MAC received)     T - Timed out
C - Config (our ID received)    M - Missing (cross-check)
X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
* - Multiple errors received    S - Standby

Domain test (level 3), Service test
Down MEP on TenGigE0/0/0/0.1 MEP-ID 3
================================================================================
St    ID MAC Address    Port    Up/Downtime   CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ----------- --------- ------ ----- -----
>     4 d46a.355c.b814 Up      00:00:24            0      0     0     0

Router#show ethernet cfm peer meps detail
Tue Nov 16 06:57:23.567 UTC
Domain test (level 3), Service test
Down MEP on TenGigE0/0/0/0.1 MEP-ID 3
================================================================================
Peer MEP-ID 4, MAC d46a.355c.b814
   CFM state: Ok, for 00:00:29
   Received CCM handling offloaded to hardware
   Port state: Up
   CCMs received: 0
     Out-of-sequence:             0
     Remote Defect received:      0
     Wrong level:                 0
     Cross-connect (wrong MAID):  0
     Wrong interval:              0
     Loop (our MAC received):     0
     Config (our ID received):    0
   Last CCM received:
     Level: 3, Version: 0, Interval: 10ms
     Sequence number: 0, MEP-ID: 4
     MAID: String: test, String: test
     Port status: Up, Interface status: Up


Router#show ethernet cfm local meps
Tue Nov 16 06:57:36.672 UTC
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
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


Domain test (level 3), Service test
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
    3 Te0/0/0/0.1 (Up)         Dn     1/0   N

Router#show ethernet cfm local meps verbose
Tue Nov 16 06:57:39.015 UTC
Domain test (level 3), Service test
Down MEP on TenGigE0/0/0/0.1 MEP-ID 3
================================================================================
  Interface state: Up      MAC address: b0c5.3cff.c080
  Peer MEPs: 1 up, 0 with errors, 0 timed out (archived)
  Cross-check errors: 0 missing, 0 unexpected

  CCM generation enabled:  Yes, 10ms (Remote Defect detected: No)
                           CCM processing offloaded to hardware
  AIS generation enabled:  No
  Sending AIS:             No
  Receiving AIS:           No
  Sending CSF:             No
  Receiving CSF:           No

  No packets sent/received
Router#
```

AIS for CFM Configuration: Examples

Example 1

This example shows how to configure Alarm Indication Signal (AIS) transmission for a CFM domain service:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ethernet cfm
```

```
RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RP0/CPU0:router(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7

RP/0/RP0/CPU0:routerconfigure
RP/0/RP0/CPU0:router(config)# ethernet cfm
RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RP0/CPU0:router(config-cfm-dmn)# service Cross_Connect_1 xconnect group XG1 p2p X1
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7
```

Example 2

This example shows how to configure AIS logging for a Connectivity Fault Management (CFM) domain service to indicate when AIS or LCK packets are received:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ethernet cfm
RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RP0/CPU0:router(config-cfm-dmn)# service S2 bridge group BG1 bridge-domain BD2
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log ais

RP/0/RP0/CPU0:routerconfigure
RP/0/RP0/CPU0:router(config)# ethernet cfm
RP/0/RP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RP0/CPU0:router(config-cfm-dmn)# service Cross_Connect_1 xconnect group XG1 p2p X1
RP/0/RP0/CPU0:router(config-cfm-dmn-svc)# log ais
```

This example shows how to configure AIS transmission on a CFM interface.

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface TenGigE 0/1/0/2
RP/0/RP0/CPU0:router(config-if)# ethernet cfm
RP/0/RP0/CPU0:router(config-if-cfm)# ais transmission up interval 1m cos 7
```

AIS for CFM Show Commands: Examples

This section includes the following examples:

show ethernet cfm interfaces ais Command: Example

This example shows how to display the information published in the Interface AIS table:

```
RP/0/RP0/CPU0:router# show ethernet cfm interfaces ais

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              D - Local port down

                               Trigger                  Transmission
                         AIS  ---------    Via    ---------------------------
Interface (State)        Dir  L Defects  Levels   L Int Last started  Packets
------------------------ ---  - -------  -------  - --- ------------ --------
TenGigE0/0/0/0 (Up)       Dn   5 RPC      6        7 1s  01:32:56 ago     5576
TenGigE0/0/0/0 (Up)       Up   0 M        2,3      5 1s  00:16:23 ago      983
```

```
TenGigE0/0/0/1 (Dn)       Up     D                 7 60s 01:02:44 ago     3764
TenGigE0/0/0/2 (Up)       Dn   0 RX       1!
```

show ethernet cfm local meps Command: Examples

Example 1: Default

This example shows how to display statistics for local maintenance end points (MEPs):

```
RP/0/RP0/CPU0:router# show ethernet cfm local meps

 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

Domain foo (level 6), Service bar
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
  100 Gi1/1/0/1 (Up)       Up     0/0   N  A       7

Domain fred (level 5), Service barney
   ID Interface (State)        Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
    2 Gi0/1/0/0 (Up)       Up     3/2   Y  RPC     6
```

Example 2: Domain Service

This example shows how to display statistics for MEPs in a domain service:

```
RP/0/RP0/CPU0:router# show ethernet cfm local meps domain foo service bar detail

Domain foo (level 6), Service bar
Down MEP on TenGigE0/0/0/1, MEP-ID 100
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)

  CCM generation enabled:  No
  AIS generation enabled:  Yes (level: 7, interval: 1s)
  Sending AIS:             Yes (started 01:32:56 ago)
  Receiving AIS:           Yes (from lower MEP, started 01:32:56 ago)

Domain fred (level 5), Service barney
Down MEP on TenGigE0/0/0/1, MEP-ID 2
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 3 up, 2 with errors, 0 timed out (archived)
  Cross-check defects: 0 missing, 0 unexpected

  CCM generation enabled:  Yes (Remote Defect detected: Yes)
  CCM defects detected:    R - Remote Defect received
                           P - Peer port down
                           C - Config (our ID received)
  AIS generation enabled:  Yes (level: 6, interval: 1s)
  Sending AIS:             Yes (to higher MEP, started 01:32:56 ago)
  Receiving AIS:           No
```

Example 4: Detail

This example shows how to display detailed statistics for MEPs in a domain service:

```
RP/0/RP0/CPU0:router# show ethernet cfm local meps detail

Domain foo (level 6), Service bar
Down MEP on TenGigE0/0/0/1, MEP-ID 100
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)

  CCM generation enabled:  No
  AIS generation enabled:  Yes (level: 7, interval: 1s)
  Sending AIS:             Yes (started 01:32:56 ago)
  Receiving AIS:           Yes (from lower MEP, started 01:32:56 ago)

Domain fred (level 5), Service barney
Down MEP on TenGigE0/0/0/1, MEP-ID 2
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 3 up, 2 with errors, 0 timed out (archived)
  Cross-check defects: 0 missing, 0 unexpected

  CCM generation enabled:  Yes (Remote Defect detected: Yes)
  CCM defects detected:    R - Remote Defect received
                           P - Peer port down
                           C - Config (our ID received)
  AIS generation enabled:  Yes (level: 6, interval: 1s)
  Sending AIS:             Yes (to higher MEP, started 01:32:56 ago)
  Receiving AIS:           No
```

show ethernet cfm local meps detail Command: Example

Use the **show ethernet cfm local meps detail** command to display MEP-related EFD status information. This example shows that EFD is triggered for MEP-ID 100:

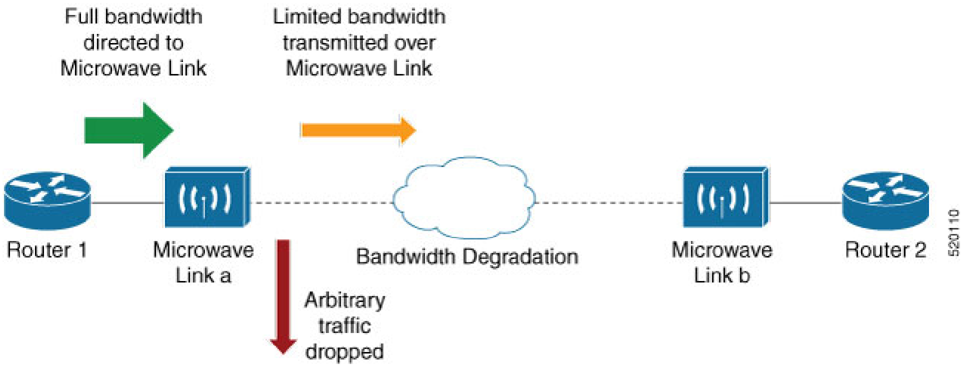
```
RP/0/RP0/CPU0:router# show ethernet cfm local meps detail

Domain foo (level 6), Service bar
Down MEP on TenGigE0/0/0/1, MEP-ID 100
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 0 up, 0 with errors, 0 timed out (archived)
  Cross-check errors: 2 missing, 0 unexpected

  CCM generation enabled:  No
  AIS generation enabled:  Yes (level: 7, interval: 1s)
  Sending AIS:             Yes (started 01:32:56 ago)
  Receiving AIS:           Yes (from lower MEP, started 01:32:56 ago)
  EFD triggered:           Yes

Domain fred (level 5), Service barney
Down MEP on TenGigE0/0/0/1, MEP-ID 2
================================================================================
  Interface state: Up      MAC address: 1122.3344.5566
  Peer MEPs: 3 up, 0 with errors, 0 timed out (archived)
  Cross-check errors: 0 missing, 0 unexpected

  CCM generation enabled:  Yes (Remote Defect detected: No)
  AIS generation enabled:  Yes (level: 6, interval: 1s)
  Sending AIS:             No
  Receiving AIS:              No
  EFD triggered:              No
```

Note You can also verify that EFD has been triggered on an interface using the **show interfaces** and **show interfaces brief** commands. When an EFD trigger has occurred, these commands will show the interface status as *up* and the line protocol state as *down*.

CFM Adaptive Bandwidth Notifications

Microwave links are used in carrier ethernet networks, because they are cheaper than laying fibre either in dense metro areas or rural locations. However, the disadvantage of microwave links is that the signal is affected by atmospheric conditions and can degrade.

Modern microwave devices support adaptive modulation schemes to prevent a complete loss of signal. This allows them to continue to operate during periods of degradation, but at a reduced bandwidth. However, to fully take advantage of this, it's necessary to convey the decrease in bandwidth to the head-end router so that appropriate actions can be taken. Otherwise, the link may become saturated and traffic dropped arbitrarily as shown in the following figure:


Full bandwidth directed to Microwave Link
Limited bandwidth transmitted over Microwave Link
Router 1
Microwave Link a
Bandwidth Degradation
Microwave Link b
Router 2
520110
Arbitrary traffic dropped

A generic solution to this is a Connectivity Fault Management (CFM) extension to send Bandwidth Notifications Messages (BNM) to Maintenance Endpoints (MEPs) on the corresponding interface on the head-end router. To be flexible in the actions taken, the choice of solution uses Embedded Event Manager (EEM) to invoke operator written TCL scripts. For information on EEM, see Embedded Event Manager, on page 64.

Bandwidth Notification Messages

The two types of messages used to notify the head-end router are:

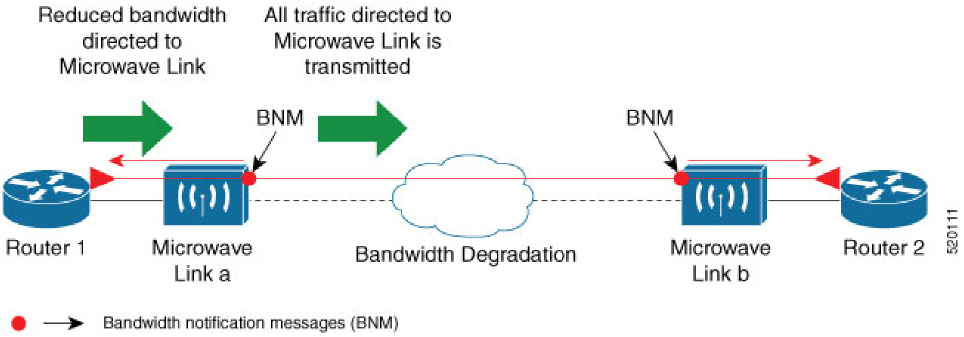
- G.8013 Bandwidth Notification Messages (G.8013 BNM)
- Cisco proprietary Bandwidth Vendor-Specific Messages (Cisco BW-VSM)

Both the message types contain the following information:

- Source MAC
- Port ID

- Maintenance Domain (MD) Level
- Transmission period
- Nominal Bandwidth
- Current Bandwidth

During signal degradation, periodic BNMs are sent to the head-end router containing the current bandwidth (sampled over a period of time) and nominal bandwidth (full bandwidth when there is no degradation). This allows the router to reduce the bandwidth directed to the link as shown in the figure below:


Reduced bandwidth directed to Microwave Link
All traffic directed to Microwave Link is transmitted
BNM
BNM
Router 1
Microwave Link a
Bandwidth Degradation
Microwave Link b
Router 2
520111
Bandwidth notification messages (BNM)

When degradation in bandwidth is detected, depending on the topology, the degradation may affect one or more paths in the network. Therefore, in more complex topologies, the head-end router may need information about links in each affected path. The BNM transmission period and a Link ID are used to differentiate between messages from the same source MAC address which refer to different links.

Restrictions for CFM Bandwidth Notifications

The list of restrictions for CFM Bandwidth Notifications is:

- Up to 200 unique BNM enabled links learnt from BNMs are supported per line card. Any BNMs for links over this limit will be discarded.

  To reset CFM BNM enabled links for the specified interfaces, use the `clear ethernet cfm interface [ <interface> ] bandwidth-notifications { all | state <state> } [ location { all | <node> } ]` command. An archive timer is used to clean up any BNM enabled links whose loss timer expired at least 24 hours ago.

- Over process restart:
  - Loss threshold, wait-to-restore, and hold-off timers are restarted. This may cause links to take longer to transition between states than they would have otherwise.
  - Archive timers are restarted. This may cause historical statistics for links to persist longer than they would have otherwise.
  - Queued events for EEM scripts which have been rate-limited are not preserved. Scripts with at least one link in DEGRADED state, or BNMs have changed over process restart, and are invoked. Rate-limit timers are restarted. This may cause scripts to be invoked when they would otherwise have been filtered by the damping or conformance-testing algorithms. If the last link returns to its nominal bandwidth within the rate-limit period but before the process restart, then the script will not be invoked after the process restart. Thus, actions taken by the script may not reflect the

(increased) latest bandwidths of any links which returned to their nominal bandwidths within the rate-limit period.

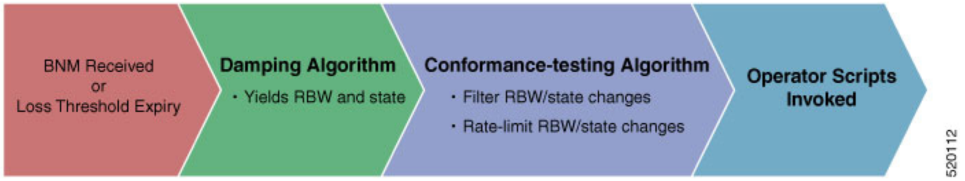
Bandwidth Reporting

Received BNMs are used to identify BNM enabled links within a Maintenance Entity Group (MEG), and should be uniquely identifiable within the MEG by Port-ID or MAC address. Each link has an associated nominal bandwidth, and a Reported Bandwidth (RBW), which are notified to the operator. The link is considered to be in OK state when the RBW is equal to the nominal bandwidth and DEGRADED if RBW is less than nominal.

Devices sending BNMs can detect changes in bandwidth many times a second. For example, changes caused by an object passing through a microwave link's line of sight. The protocol for sending BNMs is designed to mitigate fluctuating current bandwidth by sampling across a 'monitoring-interval' and applying basic damping to degradation events. To help mitigate this further, a damping algorithm is used. This algorithm is applied on the receiving device, and is distinct from any damping performed by the sender. For more information on this, see Damping Algorithm, on page 62.

An operator may be interested in more than one BNM enabled link, and needs the ability to register on a set of BNM enabled links which affect the path to a node in the network. To do this, the state and RBW for each link of interest are put into a conformance testing algorithm, which both filters and rate-limits changes to publish events notifying the operator only of significant changes. For more information on this, see Conformance Testing Algorithm, on page 64.

The following diagram shows how a received BNM flows through the damping and conformance testing algorithm to invoke operator scripts:



Note

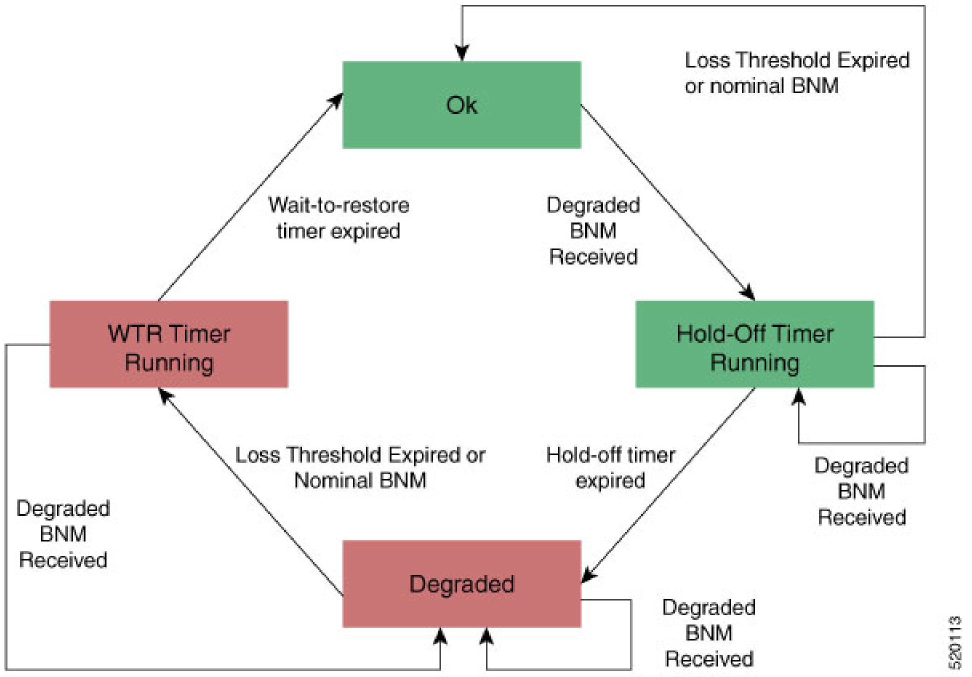
- Port ID takes precedence over MAC address. This means that BNMs with same port ID but different MAC addresses are counted as same BNMs.
- If BNM reported bandwidth is equal to the threshold, then EEM will not be invoked.
- If a degraded link having bandwidth higher than the threshold receives BNM with bandwidth less than the threshold, it doesn't wait for the hold-off timer and instantly changes the bandwidth by invoking EEM script.

Damping Algorithm

A damping algorithm is applied to each unique BNM enabled link for which BNMs are received. The table below describes the timers used for this purpose:

Timers	Description
loss threshold (in packet numbers)	This timer handles the case when BNMs stop being received. This timer is (re)started whenever any BNM is received for the link. The value is equal to the expected period between BNMs (as indicated by the last BNM) multiplied by the configured loss threshold. When the loss threshold timer expires, as the link may have changed or been removed entirely, bandwidth information is no longer available, therefore the link is considered to have been restored to its previously notified nominal bandwidth.
hold-off (in seconds)	This timer is used to damp transient transitions from OK to DEGRADED state. It is started when the first degraded BNM is received, and is stopped if the loss threshold timer expires or the current bandwidth returns to the nominal bandwidth. If the timer expires, then the BNM enabled link enters DEGRADED state. The value of this timer is configurable. If it is zero, then the link immediately enters degraded state and the timer is not started.
wait-to-restore (WTR, in seconds)	This timer is used to damp transient transitions from DEGRADED to OK state. It is started when a BNM Enabled Link is in DEGRADED state and either the loss threshold timer expires or a BNM is received that indicates the current bandwidth has returned to the nominal bandwidth. If a degraded BNM is received while the timer is running then it is stopped and the BNM Enabled Link remains in DEGRADED state. If this timer expires then the link returns to OK state.

The following internal state transition diagram shows how damping algorithm takes place:



Conformance Testing Algorithm

The conformance testing algorithm comprises of two parts:

1. Filtering bandwidth changes.

   Filtering is done so that events are published whenever either:

   - Any link which was in OK state or had a RBW more than or equal to the specified threshold, has transitioned to DEGRADED state and has a RBW less than the specified threshold.
   - Any link which was in DEGRADED state and had a RBW less than the specified threshold, is still in DEGRADED state and has a RBW less than the specified threshold, but the old and new RBWs are different.
   - Any link which was in DEGRADED state and had a RBW less than the specified threshold, has transitioned to OK state or has a RBW more than or equal to the specified threshold.

2. Rate-limiting bandwidth changes

   Rate-limiting is done by only publishing events at most once within any rate-limit period. If there is a change in bandwidth (which passes the filter) within this rate-limit period, a timer is started to expire at the end of the period. Upon timer expiry, an event is published which reflects the latest state and bandwidth of all links of interest which are in DEGRADED state.

Embedded Event Manager

The Embedded Event Manager (EEM) consists of an EEM server that monitors various real-time events in the system using programs called Event Detectors (EDs) and triggers registered policies (for example, TCLscripts) to run. The EEM supports at least 200 script registrations.

Typical actions taken in response to signal degradation events include:

- Signaling to G.8032 to switch some flows to alternative paths
- Modifying QoS configuration to adjust traffic shaping to the new bandwidth
- Adjusting IGP metrics to switch some traffic to an alternative path

The following variables can be queried within the TCL script:

EEM Variables	Comment
`interface, level, direction`	Identify the MEP in the registration
`min_reported_bandwidth`	Minimum reported bandwidth across all links in the registration that are currently in DEGRADED state, and below the specified threshold
`bnm_enabled_links [{ MAC address | Port ID }]`	Array of BNM enabled links, with each one containing the following elements: • `reported_bw`: Reported Bandwidth • `nominal_bw`: Nominal BW in last BNM
`event_type`	Either 'DEGRADED' or 'OK' DEGRADED indicates that at least one BNM enabled link in the registration is in DEGRADED state with a reported bandwidth less than the threshold. This means that the `event_type` could be 'OK' if all BNM enabled links in the registration which are in DEGRADED state have a reported bandwidth greater than or equal to the threshold.

The command for EEM TCL scripts registering for CFM Bandwidth Notification events is `interface <interface name> level <level> direction <direction> {mac-addresses { <addr1> [, …, <addr20>] } | port-ids { <id1> [, …, <id20>] } threshold <bandwidth> [ ratelimit <time> ]`.

To configure EEM, use the following commands:

```
event manager directory user policy disk0:/
event manager directory user library disk0:/
event manager policy EEMscript7.tcl username root persist-time 3600
aaa authorization eventmanager default local
```

Individual scripts located in the specified directory can then be configured with:

```
event manager policy <script_name> username lab persist-time <time>
```

Event Publishing

CFM publishes events for a given EEM registration after applying the damping and conformance testing algorithms as described in Damping Algorithm, on page 62 and Conformance Testing Algorithm, on page 64 respectively. The set of BNM Enabled Links published in an event are those in DEGRADED state and whose RBW is less than the specified threshold.

Configure CFM Bandwidth Notifications

Use the following steps to configure CFM bandwidth notifications:

- Configure a CFM domain at the level BNMs are expected to be received at, and a CFM service in the direction (either up or down-MEPs) the BNMs are expected to be received.
- Configure a CFM MEP on the interface expected to receive BNMs in the domain and service above.

Configuration consists of two parts:

- Configuring global CFM. This is similar to Continuity Check Message (CCM) and other CFM configurations.

  Global CFM configuration:

  ```
  ethernet cfm
  domain DM1 level 2 id null
    service SR1 down-meps
    !
  !
  domain dom1 level 1
    service ser1 down-meps
    !
  !
  ```

- Configuration related to CFM-BNMs under interfaces. This is optional and used for changing default values.

Interface configuration:

```
Interface TenGigE0/0/1/1
ethernet cfm
  mep domain DM1 service SR1 mep-id 3001
  !
  bandwidth-notifications
   hold-off 0
   wait-to-restore 60
   loss-threshold 10
   log changes
  !
!
l2transport
!
!
interface TenGigE0/0/0/3
ethernet cfm
  mep domain dom1 service ser1 mep-id 11
  !
  bandwidth-notifications
  hold-off 10
   wait-to-restore 40
   log changes
  !
!
l2transport
!
!
```

Running Configuration

```
RP/0/RP0/CPU0:router#show running-configuration
!! IOS XR Configuration 7.1.1.104I
```

```
!! Last configuration change at Mon Jun 24 21:26:46 2019 by root
!
hostname R2_cXR
logging console debugging
logging buffered 125000000
event manager directory user policy harddisk:/tcl/
event manager directory user library harddisk:/tcl/
event manager policy EEMmac_lev1.tcl username root persist-time 3600
event manager policy EEMport_lev1.tcl username root persist-time 3600
aaa authorization exec default local group tacacs+
aaa authorization eventmanager default local
!
ethernet cfm
 domain DM0 level 1 id null
  service SR0 down-meps
   continuity-check interval 1m
   mep crosscheck
    mep-id 1003
   !
   ais transmission interval 1s cos 4
   log ais
   log continuity-check errors
   log crosscheck errors
   log continuity-check mep changes
  !
 !
 domain DM1 level 2 id null
  service SR1 down-meps id number 1
   continuity-check interval 1m
   mep crosscheck
    mep-id 431
   !
   ais transmission interval 1m
   log ais
   log continuity-check errors
   log crosscheck errors
   log continuity-check mep changes
  !
 domain dom1 level 3 id string domain3
  service ser1 xconnect group XG1 p2p XC1 id number 2300
   mip auto-create all
   continuity-check interval 1m
   mep crosscheck
    mep-id 2030
   !
interface Loopback0
 ipv4 address 30.30.30.30 255.255.255.255
!
interface MgmtEth0/RSP0/CPU0/0
 ipv4 address 5.18.9.102 255.255.0.0
!
interface MgmtEth0/RSP0/CPU0/1
 shutdown
!
interface TenGigE0/0/0/0
 shutdown
!
interface TenGigE0/0/0/3.1 l2transport
 encapsulation dot1q 6
 ethernet cfm
  mep domain DM1 service SR1 mep-id 231
  !
  bandwidth-notifications
   hold-off 50
   wait-to-restore 50
   loss-threshold 100
   log changes
  !
```

Verification

```
RP/0/RP0/CPU0:router#show ethernet cfm interfaces bandwidth-notifications detail
BNM Enabled Links at Level 3 (Down MEP) for GigabitEthernet/1
    MAC Address 000a.000a.000a
      State (OK):
        Nominal Bandwidth:                                    3000 Mbps
        Reported Bandwidth:                                   1000 Mbps
        Elapsed time in this state:                         00:00:13.000
        Transitions into degraded state:                           5000
        Hold-off:                                         111s remaining
      Last BNM received 00:00:10 ago
        Nominal Bandwidth:                                    1000 Mbps
        Current Bandwidth:                                    2000 Mbps
        Interval:                                                   10s
        Packet-type:                                        Cisco BW-VSM
      Packets received:                                           20000

    Port ID 7 (MAC Address 000c.000c.000c)
      State (DEGRADED):
        Nominal Bandwidth:                                    6000 Mbps
        Reported Bandwidth:                                   2000 Mbps
        Elapsed time in this state:                         00:00:39.000
        Transitions into degraded state:                          10000
        Wait-to-restore:                                  111s remaining
      Last BNM received 00:00:33 ago
        Nominal Bandwidth:                                    2000 Mbps
        Current Bandwidth:                                    4000 Mbps
        Interval:                                                  1min
        Packet-type:                                        Cisco BW-VSM
      Packets received:                                           40000
```

CFM Over Bundles

CFM over bundle supports the following:

- CFM Maintenance Points—Up Maintenance-association End Points (MEP), Down MEP, and MIP, which includes L2 bundle main and sub-interfaces.
- CCM interval of 100 microsecond, 1second, 10 seconds, and 1 minute. CCM interval of 10 minutes is supported only in the versions earlier than IOS XR 7.3.2.
- RP OIR/VM reload, without impacting learned CFM peer MEPs.
- Process restart without impacting CFM sessions.
- CFM MEPs on bundle interfaces as software-offloaded-MEPs with all possible rewrite and encapsulation combinations supported by L2 sub-interfaces.
- CCM learning on MIP over bundle interfaces. CCM database learning supports investigation of one CCM out of 50 that goes over MIP.
- Static and dynamic Remote MEPs.

Restrictions for Configuration of CFM on Bundles

Following are the restrictions for configuring CFM:

- Only Layer 2 bundle Ethernet interfaces and sub-interfaces are supported except for those matching the VLAN tag `any`.
- CCM interval of 3.3 milliseconds and 10 milliseconds are not supported.
- CCM interval of 10 minutes is not supported from IOS XR 7.3.2.
- Supports 5000 pps rates of CCM traffic for bundle interfaces. For example, for CCM interval of 100 milliseconds, the number of MEPs can be 500.
- Ethernet CFM is not supported with MEP that are configured on default and untagged encapsulated sub-interfaces that are part of a single physical interface.

CFM with SAT and EDPL

CFM can run along with SAT (Service Activation Test) session on the same interface. Both works independent of each other.

However, other OAM sessions like SLM and DMM will go down during the SAT session. They get restored once the SAT session is completed. This is expected as per requirements.

Limitations and Restrictions

- SAT session works similar to MD-level 7 session. So, CFM sessions, on same interface, will have to be at levels lower than 7, i.e 0 to 6.

Example:

The below setup is an example:

```
Interface 1           ---------------------         Interface 2
         CFM (MDL 0 to 6)      ---------------------         CFM (MDL 0 to 6)
         DMM/SLM               ----------------              DMM/SLM
         SAT                   --------------                EDPL (with DestMac)
```

Note

- DMM/SLM goes down when SAT is active. They get restored once SAT session is completed.
- Ethernet Data Plane Loopback functionality (EDPL) does not support multicast destination MAC address packets for NCS 5700 line cards. So, it is recommended to use EDPL on peer node with filter - Destination_MAC (same as the destination of the SAT session).
- CCM have multicast destination MAC(0180.c200.003x).

CFM over Static L2VPN and LSP with Single-Pass GRE Tunnel

Table 3: Feature History Table

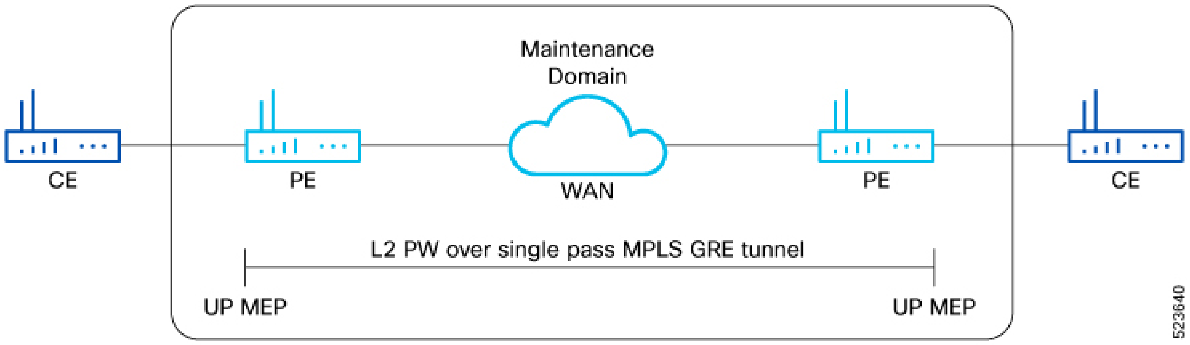
Feature Name	Release Information	Description
CFM over Static L2VPN and LSP with Single-Pass GRE Tunnel	Release 24.1.1	Introduced in this release on: NCS 5500 fixed port routers (select variants only*); NCS 5500 modular routers (select variants only*) By activating Connectivity Fault Management (CFM) when using GRE tunnel as the underlying transport mechanism, you can now monitor and isolate faults within a maintenance domain. CFM is now available over static L2VPN and LSP with single-pass GRE tunnels on your PE routers. It helps check if L2VPN services are working and possibly take corrective actions if they aren't. This capability enhances tunnel health monitoring and fault identification across the pseudowire (PW) tunnel between edge routers.

Using CFM over L2 pseudowire on single-pass MPLS GRE tunnels, you can now identify connectivity issues in a tunnel between PE routers. Previously, the support for CFM over MPLS GRE tunnels wasn't available. To know more about CFM functionality and its configuration, see Ethernet CFM, on page 15.

Topology

Let's understand how the CFM capability works over L2 PW tunnels using a sample topology.

Figure 10: Sample Topology



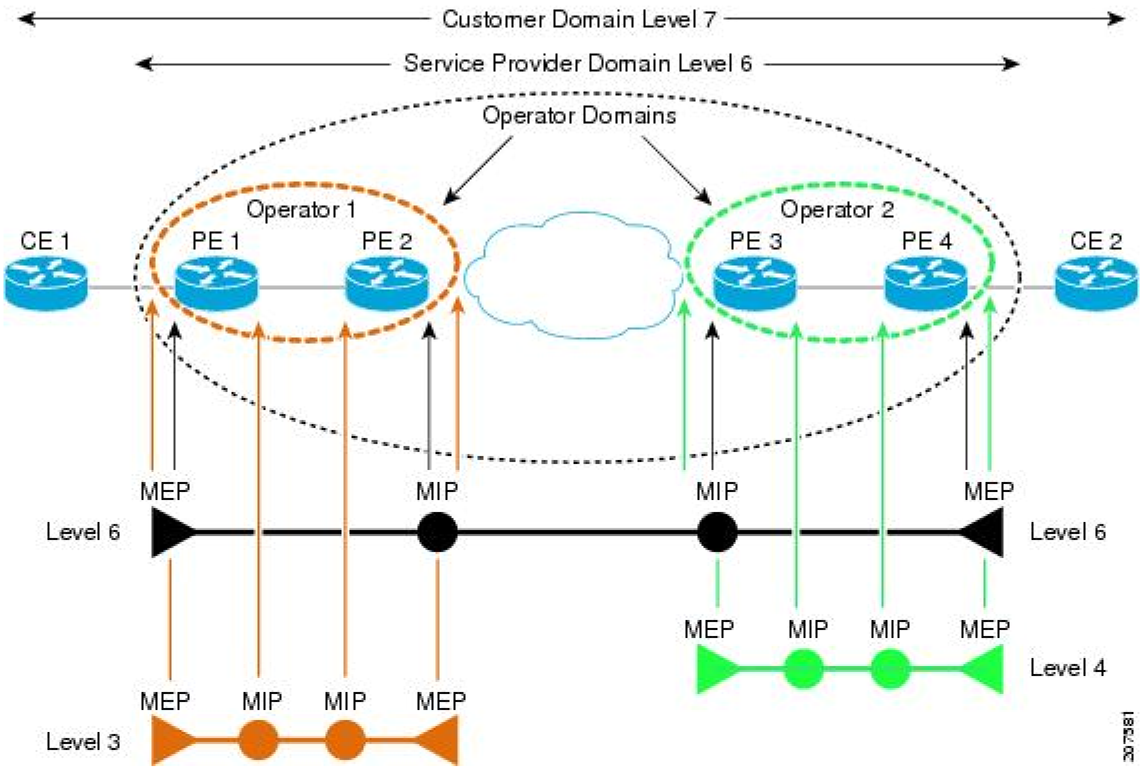
In this topology,

- The provider edge (PE) routers connect to customer on-premise equipment (CE).
- Two UP Maintenance End Points (MEPs) at the edge of the maintenance domain connect via an L2 PW on a single-pass MPLS GRE tunnel with a label-switched path (LSP).

  For more information on MEPs, see Maintenance Points.

The preceding topology is a subsection of the different CFM maintenance domains configured across a network, as shown in the following figure.

Figure 11: Different CFM Maintenance Domains Across a Network



CFM is configured with UP-UP connections at the PE routers through WAN. CFM PDUs are based on the Y.1731 standard. The PDUs travel through the L2 PW over an MPLSoGRE tunnel between the UP MEPs. For more information, see MEP and CFM Processing Overview and CFM Protocol Messages, on page 22.

The MEPs send Continuity Check Messages (CCMs) periodically to verify the connection between the PE routers. If the MEP detects a fault, it sends an Alarm Indication Signal (AIS) message. The AIS messages are sent via multicast similar to CCMs. However, unlike CCMs, AIS messages are sent in the direction away from the peer remote MEPs and towards the higher domain level than the sender MEP. As a result, the faults detected in one Maintenance Association (MA) are notified to the MA at the higher level. An example use case is to notify a customer network of the faults in the service provider network.

An interface without MEPs can also send AIS messages if it has a Maintenance Intermediate Point (MIP). In this case, the only fault that triggers the AIS message is the interface state going down. The MIPs forward the AIS messages to the bridge at their level without responding to the message. On receiving the AIS message at the bridge, the MEPs begin sending the AIS message to the next higher domain level. The process repeats until the message propagates to the highest-level domain. For more information on MIPs, see MIP Creation, on page 19.

Restrictions for CFM over Static L2VPN and LSP with Single-Pass GRE Tunnel

- Support for CFM over L2oMPLS with two-pass GRE tunnel isn't available.
- Y.1731 performance monitoring isn't available for this feature.
- Only Table 4: Supported Platform Variants support this feature.

Supported Platform Variants for CFM over Static L2VPN and LSP with Single-Pass GRE Tunnel

The following table lists the platform variants that support this feature.

Table 4: Supported Platform Variants

NCS 5500 fixed port routers	NCS-55A1-24H NCS-55A1-36H-S NCS-55A1-36H-SE-S NCS-55A1-48Q6H NCS-55A1-24Q6H-S NCS-55A1-24Q6H-SS
NCS 5500 modular routers	NC55-36x100G-A-SE

Configure CFM over Static L2VPN and LSP with Single-Pass GRE Tunnel

This section describes how to configure CFM over MPLS GRE tunnel, while also considering the following aspects:

- GRE tunnel is configured in a single-pass encapsulation mode.
- Policy Based Routing (PBR) decapsulation is used for single-pass GRE decapsulation.
- L2VPN “Control word” is supported along with the single-pass GRE tunnel.
- Single-pass PBR decapsulation configuration is used for GRE decapsulation.

Configuration Example

PE1:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#class-map type traffic match-all test_gre1
RP/0/RP0/CPU0:ios(config-cmap)#match protocol gre
RP/0/RP0/CPU0:ios(config-cmap)#match source-address ipv4 198.51.100.101 255.255.255.0
RP/0/RP0/CPU0:ios(config-cmap)# end-class-map
RP/0/RP0/CPU0:ios(config)#policy-map type pbr P1-test
RP/0/RP0/CPU0:ios(config-pmap)#class type traffic test_gre1
RP/0/RP0/CPU0:ios(config-pmap-c)#decapsulate gre
RP/0/RP0/CPU0:ios(config-pmap-c)#exit
RP/0/RP0/CPU0:ios(config-pmap)#class type traffic class-default
RP/0/RP0/CPU0:ios(config-pmap-c)#exit
RP/0/RP0/CPU0:ios(config-pmap)#end-policy-map
RP/0/RP0/CPU0:ios(config)#vrf-policy
```

```
RP/0/RP0/CPU0:ios(config-vrf-policy)#vrf default address-family ipv4 policy type pbr input
 P1-test
RP/0/RP0/CPU0:ios(config-vrf-policy)#exit
RP/0/RP0/CPU0:ios(config)#interface Loopback0
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 198.51.100.100 255.255.255.0
RP/0/RP0/CPU0:ios(config-if)#exit
RP/0/RP0/CPU0:ios(config)#interface Loopback11
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 10.11.11.11 255.0.0.0
RP/0/RP0/CPU0:ios(config-if)#exit
RP/0/RP0/CPU0:ios(config)#interface Bundle-ether100
RP/0/RP0/CPU0:ios(config-if)#interface Bundle-ether100.1 l2transport
RP/0/RP0/CPU0:ios(config-subif)#encapsulation dot1q 1
RP/0/RP0/CPU0:ios(config-subif)#ethernet cfm
RP/0/RP0/CPU0:ios(config-if-cfm)#mep domain UP6 service s61 mep-id 1
RP/0/RP0/CPU0:ios(config-if-cfm-mep)#exit
RP/0/RP0/CPU0:ios(config-if-cfm)#interface TenGigE0/0/0/16/0
RP/0/RP0/CPU0:ios(config-if)#bundle id 100
RP/0/RP0/CPU0:ios(config-if)#exit
RP/0/RP0/CPU0:ios(config)#interface TenGigE0/0/0/17/1
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 10.18.18.1 255.0.0.0
RP/0/RP0/CPU0:ios(config-if)#exit
RP/0/RP0/CPU0:ios(config)#interface tunnel-ip100
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 198.51.100.111 255.255.255.0
RP/0/RP0/CPU0:ios(config-if)#tunnel mode gre ipv4 encap
RP/0/RP0/CPU0:ios(config-if)#tunnel source 198.51.100.100
RP/0/RP0/CPU0:ios(config-if)#tunnel destination 198.51.100.101
RP/0/RP0/CPU0:ios(config-if)#exit
RP/0/RP0/CPU0:ios(config)#router static
RP/0/RP0/CPU0:ios(config-static)#address-family ipv4 unicast
RP/0/RP0/CPU0:ios(config-static-afi)#10.22.22.22/32 tunnel-ip100
RP/0/RP0/CPU0:ios(config-static-afi)#exit
RP/0/RP0/CPU0:ios(config-static)#exit
RP/0/RP0/CPU0:ios(config)#router bgp 100
RP/0/RP0/CPU0:ios(config-bgp)#nsr
RP/0/RP0/CPU0:ios(config-bgp)#bgp router-id 198.51.100.100
RP/0/RP0/CPU0:ios(config-bgp)#bgp graceful-restart
RP/0/RP0/CPU0:ios(config-bgp)#bgp log neighbor changes detail
RP/0/RP0/CPU0:ios(config-bgp)#address-family ipv4 unicast
RP/0/RP0/CPU0:ios(config-bgp-af)#maximum-paths ebgp 64
RP/0/RP0/CPU0:ios(config-bgp-af)#neighbor 198.51.100.101
RP/0/RP0/CPU0:ios(config-bgp-nbr)#remote-as 300
RP/0/RP0/CPU0:ios(config-bgp-nbr)#ebgp-multihop 10
RP/0/RP0/CPU0:ios(config-bgp-nbr)#update-source Loopback0
RP/0/RP0/CPU0:ios(config-bgp-nbr)#address-family ipv4 unicast
RP/0/RP0/CPU0:ios(config-bgp-nbr-af)#next-hop-self
RP/0/RP0/CPU0:ios(config-bgp-nbr-af)#route-policy pass-all in
RP/0/RP0/CPU0:ios(config-bgp-nbr-af)#route-policy pass-all out
RP/0/RP0/CPU0:ios(config-bgp-nbr-af)#exit
RP/0/RP0/CPU0:ios(config-bgp-nbr)#exit
RP/0/RP0/CPU0:ios(config-bgp)#l2vpn
RP/0/RP0/CPU0:ios(config-l2vpn)#pw-class controlword
RP/0/RP0/CPU0:ios(config-l2vpn-pwc)#encapsulation mpls
RP/0/RP0/CPU0:ios(config-l2vpn-pwc-mpls)#control-word
RP/0/RP0/CPU0:ios(config-l2vpn-pwc-mpls)#exit
RP/0/RP0/CPU0:ios(config-l2vpn-pwc)#exit
RP/0/RP0/CPU0:ios(config-l2vpn)#xconnect group 1
RP/0/RP0/CPU0:ios(config-l2vpn-xc)#p2p 1
RP/0/RP0/CPU0:ios(config-l2vpn-xc-p2p)#interface Bundle-ether100.1
RP/0/RP0/CPU0:ios(config-l2vpn-xc-p2p)#neighbor ipv4 10.22.22.22 pw-id 1
RP/0/RP0/CPU0:ios(config-l2vpn-xc-p2p-pw)#mpls static label local 25011 remote 25022
RP/0/RP0/CPU0:ios(config-l2vpn-xc-p2p-pw)#pw-class controlword
RP/0/RP0/CPU0:ios(config-l2vpn-xc-p2p-pw)#exit
```

```
RP/0/RP0/CPU0:ios(config-l2vpn-xc-p2p)#exit
RP/0/RP0/CPU0:ios(config-l2vpn-xc)#exit
RP/0/RP0/CPU0:ios(config-l2vpn)#exit
RP/0/RP0/CPU0:ios(config)#mpls static
RP/0/RP0/CPU0:ios(config-mpls-static)#interface TenGigE0/0/0/17/1
RP/0/RP0/CPU0:ios(config-mpls-static)#address-family ipv4 unicast
RP/0/RP0/CPU0:ios(config-mpls-static-af)#exit
RP/0/RP0/CPU0:ios(config-mpls-static)#lsp v4_gre_mpls_1
RP/0/RP0/CPU0:ios(config-mpls-static-lsp)#in-label 24022 allocate per-prefix 10.22.22.22/32
RP/0/RP0/CPU0:ios(config-mpls-static-lsp)#forward
RP/0/RP0/CPU0:ios(config-mpls-static-lsp-fwd)#path 1 nexthop tunnel-ip100 out-label pop
RP/0/RP0/CPU0:ios(config-mpls-static-lsp-fwd)#exit
RP/0/RP0/CPU0:ios(config-mpls-static-lsp)#exit
RP/0/RP0/CPU0:ios(config-mpls-static)#exit
RP/0/RP0/CPU0:ios(config)#ethernet cfm
RP/0/RP0/CPU0:ios(config-cfm)#domain UP6 level 6 id null
RP/0/RP0/CPU0:ios(config-cfm-dmn)#service s6 xconnect group 1 p2p 1 id number 6
RP/0/RP0/CPU0:ios(config-cfm-dmn-svc)#continuity-check interval 100ms
RP/0/RP0/CPU0:ios(config-cfm-dmn-svc)#mep crosscheck
RP/0/RP0/CPU0:ios(config-cfm-xcheck)#mep-id 4001
RP/0/RP0/CPU0:ios(config-cfm-xcheck)#exit
RP/0/RP0/CPU0:ios(config-cfm-dmn-svc)#end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

PE2:

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#class-map type traffic match-all test_gre1
RP/0/RP0/CPU0:ios(config-cmap)#match protocol gre
RP/0/RP0/CPU0:ios(config-cmap)#match source-address ipv4 198.51.100.100 255.255.255.0
RP/0/RP0/CPU0:ios(config-cmap)# end-class-map
RP/0/RP0/CPU0:ios(config)#policy-map type pbr P1-test
RP/0/RP0/CPU0:ios(config-pmap)#class type traffic test_gre1
RP/0/RP0/CPU0:ios(config-pmap-c)#decapsulate gre
RP/0/RP0/CPU0:ios(config-pmap-c)#exit
RP/0/RP0/CPU0:ios(config-pmap)#class type traffic class-default
RP/0/RP0/CPU0:ios(config-pmap-c)#exit
RP/0/RP0/CPU0:ios(config-pmap)#end-policy-map
RP/0/RP0/CPU0:ios(config)#vrf-policy
RP/0/RP0/CPU0:ios(config-vrf-policy)#vrf default address-family ipv4 policy type pbr input
 P1-test
RP/0/RP0/CPU0:ios(config-vrf-policy)#exit
RP/0/RP0/CPU0:ios(config)#interface Loopback0
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 198.51.100.101 255.255.255.0
RP/0/RP0/CPU0:ios(config-if)#exit
RP/0/RP0/CPU0:ios(config)#interface Loopback22
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 10.22.22.22 255.255.255.255
RP/0/RP0/CPU0:ios(config-if)#exit
RP/0/RP0/CPU0:ios(config)#interface tunnel-ip100
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 198.51.100.112 255.255.255.0
RP/0/RP0/CPU0:ios(config-if)#tunnel mode gre ipv4 encap
RP/0/RP0/CPU0:ios(config-if)#tunnel source 198.51.100.101
RP/0/RP0/CPU0:ios(config-if)#tunnel destination 198.51.100.100
RP/0/RP0/CPU0:ios(config-if)#exit
RP/0/RP0/CPU0:ios(config)#interface TenGigE0/0/0/24
RP/0/RP0/CPU0:ios(config-if)#ipv4 address 10.18.18.2 255.0.0.0
RP/0/RP0/CPU0:ios(config-if)#exit
RP/0/RP0/CPU0:ios(config)#interface Bundle-ether100
RP/0/RP0/CPU0:ios(config-if)#interface Bundle-ether100.1 l2transport
RP/0/RP0/CPU0:ios(config-subif)#encapsulation dot1q 1
RP/0/RP0/CPU0:ios(config-subif)#ethernet cfm
RP/0/RP0/CPU0:ios(config-if-cfm)#mep domain UP6 service s6 mep-id 4001
RP/0/RP0/CPU0:ios(config-if-cfm-mep)#exit
```

```
RP/0/RP0/CPU0:ios(config-if-cfm)#interface TenGigE0/0/0/26
RP/0/RP0/CPU0:ios(config-if)#bundle id 100
RP/0/RP0/CPU0:ios(config-if)#exit
RP/0/RP0/CPU0:ios(config)#router static
RP/0/RP0/CPU0:ios(config-static)#address-family ipv4 unicast
RP/0/RP0/CPU0:ios(config-static-afi)#10.10.10.11/32 tunnel-ip100
RP/0/RP0/CPU0:ios(config-static-afi)#exit
RP/0/RP0/CPU0:ios(config-static)#exit
RP/0/RP0/CPU0:ios(config)#router bgp 300
RP/0/RP0/CPU0:ios(config-bgp)#nsr
RP/0/RP0/CPU0:ios(config-bgp)#bgp router-id 198.51.100.101
RP/0/RP0/CPU0:ios(config-bgp)#bgp graceful-restart
RP/0/RP0/CPU0:ios(config-bgp)#bgp log neighbor changes detail
RP/0/RP0/CPU0:ios(config-bgp)#address-family ipv4 unicast
RP/0/RP0/CPU0:ios(config-bgp-af)#maximum-paths ebgp 64
RP/0/RP0/CPU0:ios(config-bgp-af)#neighbor 198.51.100.100
RP/0/RP0/CPU0:ios(config-bgp-nbr)#remote-as 100
RP/0/RP0/CPU0:ios(config-bgp-nbr)#ebgp-multihop 10
RP/0/RP0/CPU0:ios(config-bgp-nbr)#update-source Loopback0
RP/0/RP0/CPU0:ios(config-bgp-nbr)#address-family ipv4 unicast
RP/0/RP0/CPU0:ios(config-bgp-nbr-af)#next-hop-self
RP/0/RP0/CPU0:ios(config-bgp-nbr-af)#route-policy pass-all in
RP/0/RP0/CPU0:ios(config-bgp-nbr-af)#route-policy pass-all out
RP/0/RP0/CPU0:ios(config-bgp-nbr-af)#exit
RP/0/RP0/CPU0:ios(config-bgp-nbr)#exit
RP/0/RP0/CPU0:ios(config-bgp)#l2vpn
RP/0/RP0/CPU0:ios(config-l2vpn)#pw-class controlword
RP/0/RP0/CPU0:ios(config-l2vpn-pwc)#encapsulation mpls
RP/0/RP0/CPU0:ios(config-l2vpn-pwc-mpls)#control-word
RP/0/RP0/CPU0:ios(config-l2vpn-pwc-mpls)#exit
RP/0/RP0/CPU0:ios(config-l2vpn-pwc)#exit
RP/0/RP0/CPU0:ios(config-l2vpn)#xconnect group 1
RP/0/RP0/CPU0:ios(config-l2vpn-xc)#p2p 1
RP/0/RP0/CPU0:ios(config-l2vpn-xc-p2p)#interface bundle-ether100.1
RP/0/RP0/CPU0:ios(config-l2vpn-xc-p2p)#neighbor ipv4 10.10.10.11 pw-id 1
RP/0/RP0/CPU0:ios(config-l2vpn-xc-p2p-pw)#mpls static label local 25022 remote 25011
RP/0/RP0/CPU0:ios(config-l2vpn-xc-p2p-pw)#pw-class controlword
RP/0/RP0/CPU0:ios(config-l2vpn-xc-p2p-pw)#exit
RP/0/RP0/CPU0:ios(config-l2vpn-xc-p2p)#exit
RP/0/RP0/CPU0:ios(config-l2vpn-xc)#exit
RP/0/RP0/CPU0:ios(config-l2vpn)#exit
RP/0/RP0/CPU0:ios(config)#ethernet cfm
RP/0/RP0/CPU0:ios(config-cfm)#domain UP6 level 6 id null
RP/0/RP0/CPU0:ios(config-cfm-dmn)#service s6 xconnect group 1 p2p 1 id number 6
RP/0/RP0/CPU0:ios(config-cfm-dmn-svc)#continuity-check interval 100ms
RP/0/RP0/CPU0:ios(config-cfm-dmn-svc)#mep crosscheck
RP/0/RP0/CPU0:ios(config-cfm-xcheck)#mep-id 1
RP/0/RP0/CPU0:ios(config-cfm-xcheck)#exit
RP/0/RP0/CPU0:ios(config-cfm)#exit
RP/0/RP0/CPU0:ios(config)#mpls static
RP/0/RP0/CPU0:ios(config-mpls-static)#interface TenGigE0/0/0/24
RP/0/RP0/CPU0:ios(config-mpls-static)#address-family ipv4 unicast
RP/0/RP0/CPU0:ios(config-mpls-static-af)#exit
RP/0/RP0/CPU0:ios(config-mpls-static)#lsp v4_gre_mpls_1
RP/0/RP0/CPU0:ios(config-mpls-static-lsp)#in-label 24011 allocate per-prefix 10.10.10.11/32
RP/0/RP0/CPU0:ios(config-mpls-static-lsp)#forward
RP/0/RP0/CPU0:ios(config-mpls-static-lsp-fwd)#path 1 nexthop tunnel-ip100 out-label pop
RP/0/RP0/CPU0:ios(config-mpls-static-lsp-fwd)#exit
RP/0/RP0/CPU0:ios(config-mpls-static-lsp)#exit
RP/0/RP0/CPU0:ios(config-mpls-static)#exit
RP/0/RP0/CPU0:ios(config-mpls)#end
Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:yes
```

Running Configuration

PE1:

```
class-map type traffic match-all test_gre1
 match protocol gre
 match source-address ipv4 198.51.100.101 255.255.255.0
 end-class-map
!
policy-map type pbr P1-test
 class type traffic test_gre1
  decapsulate gre
 !
 class type traffic class-default
 !
 end-policy-map
!
vrf-policy
 vrf default address-family ipv4 policy type pbr input P1-test
!
interface Loopback0
 ipv4 address 198.51.100.100 255.255.255.0
!
interface Loopback11
 ipv4 address 10.11.11.11 255.0.0.0
!
interface Bundle-ether100
interface Bundle-ether100.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain UP6 service s61 mep-id 1
!
interface TenGigE0/0/0/16/0
 bundle id 100
!
interface TenGigE0/0/0/17/1
 ipv4 address 10.18.18.1 255.0.0.0
!
interface tunnel-ip100
 ipv4 address 198.51.100.111 255.255.255.0
 tunnel mode gre ipv4 encap
 tunnel source 198.51.100.100
 tunnel destination 198.51.100.101
!
router static
 address-family ipv4 unicast
  10.22.22.22/32 tunnel-ip100
 !
!
router bgp 100
 nsr
 bgp router-id 198.51.100.100
 bgp graceful-restart
 bgp log neighbor changes detail
 address-family ipv4 unicast
   maximum-paths ebgp 64
 neighbor 198.51.100.101
   remote-as 300
   ebgp-multihop 10
   update-source Loopback0
   address-family ipv4 unicast
    next-hop-self
   route-policy pass-all in
   route-policy pass-all out
  !
```

```
!
l2vpn
  pw-class controlword
    encapsulation mpls
     control-word
    !
 !
 xconnect group 1
  p2p 1
   interface Bundle-ether100.1
   neighbor ipv4 10.22.22.22 pw-id 1
    mpls static label local 25011 remote 25022
    pw-class controlword
   !
  !
 !
!
mpls static
 interface TenGigE0/0/0/17/1
 address-family ipv4 unicast
 !
  lsp v4_gre_mpls_1
  in-label 24022 allocate per-prefix 10.22.22.22/32
  forward
   path 1 nexthop tunnel-ip100 out-label pop
  !
 !
!
ethernet cfm
domain UP6 level 6 id null
  service s6 xconnect group 1 p2p 1 id number 6
   continuity-check interval 100ms
   mep crosscheck
    mep-id 4001
!
```

PE2:

```
 class-map type traffic match-all test_gre1
 match protocol gre
 match source-address ipv4 198.51.100.100 255.255.255.0
 end-class-map
!
policy-map type pbr P1-test
 class type traffic test_gre1
  decapsulate gre
 !
 class type traffic class-default
 !
 end-policy-map
!
vrf-policy
 vrf default address-family ipv4 policy type pbr input P1-test
!
interface Loopback0
 ipv4 address 198.51.100.101 255.255.255.0
!
interface Loopback22
 ipv4 address 10.22.22.22 255.255.255.255
!
interface tunnel-ip100
 ipv4 address 198.51.100.112 255.255.255.0
 tunnel mode gre ipv4 encap
 tunnel source 198.51.100.101
 tunnel destination 198.51.100.100
```

```
!
interface TenGigE0/0/0/24
 ipv4 address 10.18.18.2 255.0.0.0
!
!
interface Bundle-ether100
!
interface Bundle-ether100.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain UP6 service s6 mep-id 4001
!
interface TenGigE0/0/0/26
bundle id 100
!
router static
 address-family ipv4 unicast
  10.10.10.11/32 tunnel-ip100
 !
!
router bgp 300
 nsr
 bgp router-id 198.51.100.101
 bgp graceful-restart
 bgp log neighbor changes detail
 address-family ipv4 unicast
  maximum-paths ebgp 64
 neighbor 198.51.100.100
  remote-as 100
  ebgp-multihop 10
  update-source Loopback0
  address-family ipv4 unicast
   next-hop-self
   route-policy pass-all in
   route-policy pass-all out
  !
!
l2vpn
  pw-class controlword
    encapsulation mpls
     control-word
    !
 !
 xconnect group 1
  p2p 1
   interface bundle-ether100.1
   neighbor ipv4 10.10.10.11 pw-id 1
    mpls static label local 25022 remote 25011
    pw-class controlword
!
ethernet cfm
domain UP6 level 6 id null
  service s6 xconnect group 1 p2p 1 id number 6
   continuity-check interval 100ms
   mep crosscheck
    mep-id 1
!
mpls static
 interface TenGigE0/0/0/24
  address-family ipv4 unicast
 !
 lsp v4_gre_mpls_1
  in-label 24011 allocate per-prefix 10.10.10.11/32
  forward
```

```
   path 1 nexthop tunnel-ip100 out-label pop
  !
!
```

Verification

This section provides some sample commands and corresponding outputs to verify the CFM configuration.

1. Check peer MEP status

```
RP/0/RP0/CPU0:ios#show ethernet cfm peer meps | utility more
Wed Jan 10 13:38:27.713 UTC
Flags:
> - Ok                          I - Wrong interval
R - Remote Defect received      V - Wrong level
L - Loop (our MAC received)     T - Timed out
C - Config (our ID received)    M - Missing (cross-check)
X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
* - Multiple errors received    S - Standby

Domain sp (level 3), Service s1
Up MEP on Bundle-Ether191.1 MEP-ID 1
================================================================================
St    ID MAC Address    Port    Up/Downtime   CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ----------- --------- ------ ----- -----
>  5001 008a.968d.54db Up      02:56:38       106011      0     1     0
```

2. Check local MEP status

```
RP/0/RP0/CPU0:ios#show ethernet cfm local meps interface Bundle-Ether191.1 verbose
Wed Jan 10 13:39:17.994 UTC
Domain sp (level 3), Service s1
Up MEP on Bundle-Ether191.1 MEP-ID 1
================================================================================
  Interface state: Up      MAC address: fcbc.cec4.e48a
  Peer MEPs: 1 up, 0 with errors, 0 timed out (archived)
  Cross-check errors: 0 missing, 0 unexpected

  CCM generation enabled:  Yes, 100ms (Remote Defect detected: No)
                           CCM processing offloaded to software
  AIS generation enabled:  Yes (level: 5, interval: 1s)
  Sending AIS:             No
  Receiving AIS:           No
  Sending CSF:             No
  Receiving CSF:           No

  Packet        Sent      Received
  ------  ----------  ---------------------------------------------------------
  CCM         106571        106514  (out of seq: 0)
  AIS              7             0
```

3. Check complete CFM configuration

```
RP/0/RP0/CPU0:ios#show ethernet cfm summary
Wed Jan 10 13:40:43.106 UTC

CFM System Summary
==================

Domains                                    2
Services                                1600
Total CCM rate (pps)                  305000
Local Meps                              1500
  Operational                           1500
    Down MEPs                              0
    Up MEPs                        1500
    Offloaded                      1500
      3.3ms                        1000
      10ms                            0
      100ms or greater              500
  Disabled (misconfiguration)         0
  Disabled (resource limit)           0
  Disabled (operational error)        0
Peer MEPs                          1500
  Operational                      1500
    Defect detected                   0
    No defect detected             1500
  Timed out                           0
MIPs                                100
Interfaces                         1500
Bridge domains/Xconnects           1500
Traceroute cache entries              0
Traceroute cache replies              0
CCM Learning database entries      1500
BNM Enabled Links                     0

CFM Summary for 0/3/CPU0
========================

Domains                               2
Services                           1600
Total CCM rate (pps)                  0
Local Meps                            0
  Operational                         0
    Down MEPs                         0
    Up MEPs                           0
    Offloaded                         0
      3.3ms                           0
      10ms                            0
      100ms or greater                0
  Disabled (misconfiguration)         0
  Disabled (resource limit)           0
  Disabled (operational error)        0
Peer MEPs                             0
  Operational                         0
    Defect detected                   0
    No defect detected                0
  Timed out                           0
MIPs                                  0
Interfaces                            0
Bridge domains/Xconnects           1500
Traceroute cache entries              0
Traceroute cache replies              0
CCM Learning database entries         0
BNM Enabled Links                     0
ISSU Role                       Primary

CFM Summary for 0/7/CPU0
========================

Domains                               2
Services                           1600
Total CCM rate (pps)             300000
Local Meps                         1000
  Operational                      1000
    Down MEPs                         0
    Up MEPs                        1000
    Offloaded                      1000
      3.3ms                        1000
      10ms                                0
      100ms or greater                    0
  Disabled (misconfiguration)             0
  Disabled (resource limit)               0
  Disabled (operational error)            0
Peer MEPs                              1000
  Operational                          1000
    Defect detected                       0
    No defect detected                 1000
  Timed out                               0
MIPs                                      0
Interfaces                             1000
Bridge domains/Xconnects               1500
Traceroute cache entries                  0
Traceroute cache replies                  0
CCM Learning database entries          1000
BNM Enabled Links                         0
ISSU Role                           Primary

CFM Summary for 0/RP0/CPU0
==========================

Domains                                   2
Services                               1600
Total CCM rate (pps)                   5000
Local Meps                              500
  Operational                           500
    Down MEPs                             0
    Up MEPs                             500
    Offloaded                           500
      3.3ms                               0
      10ms                                0
      100ms or greater                  500
  Disabled (misconfiguration)             0
  Disabled (resource limit)               0
  Disabled (operational error)            0
Peer MEPs                               500
  Operational                           500
    Defect detected                       0
    No defect detected                  500
  Timed out                               0
MIPs                                    100
Interfaces                              500
Bridge domains/Xconnects                500
Traceroute cache entries                  0
Traceroute cache replies                  0
CCM Learning database entries           500
BNM Enabled Links                         0
ISSU Role                           Primary
```

4. Ping and Traceroute

```
RP/0/RP0/CPU0:ios#ping ethernet cfm domain sp service s1 mep-id 5001 source interface
Bundle-Ether191.1
Wed Jan 10 13:41:59.562 UTC
Type escape sequence to abort.
Sending 5 CFM Loopbacks, timeout is 2 seconds -
Domain sp (level 3), Service s1
Source: MEP ID 1, interface Bundle-Ether191.1
Target: 008a.968d.54db (MEP ID 5001):
  Running (5s) ...
Success rate is 100.0 percent (5/5), round-trip min/avg/max = 1/1/3 ms
Out-of-sequence: 0.0 percent (0/5)
Bad data: 0.0 percent (0/5)
Received packet rate: 1.3 pps
```

```
RP/0/RP0/CPU0:ios#traceroute ethernet cfm domain sp service s1 mep-id 5001 source interface
 Bundle-Ether191.1
Wed Jan 10 13:42:13.129 UTC
Type escape sequence to return to prompt.

Traceroutes in domain sp (level 3), service s1
Source: MEP-ID 1, interface Bundle-Ether191.1
================================================================================
Traceroute at 2024-01-10 13:42:13 to 008a.968d.54db,
TTL 64, Trans ID 677741245:

  Running (7s) ...

Hop Hostname/Last            Ingress MAC/name       Egress MAC/Name            Relay
--- ------------------------ ---------------------- ---------------------- ----------
  1 R1-5508                  fcbc.cec4.e48a [Ok]     FDB 0000-fcbc.cec4.e48a   BE191.1

  2 R3                       008a.968d.54db [Ok]     Hit R1-5508               BE391.1
            MEP
Replies dropped: 0
```

Y.1731 Performance Monitoring

Table 5: Feature History Table

Feature Name	Release	Description
Cisco NC57 Native Mode: Y.1731 Loss and Delay Measurement	Release 7.3.1	This feature is now supported on routers that have the Cisco NC57 line cards installed and operate in the native mode.

Y.1731 Performance Monitoring (PM) provides a standard Ethernet PM function that includes measurement of Ethernet frame delay, frame delay variation, frame loss, and frame throughput measurements. This is specified by the ITU-T Y-1731 standard and interpreted by the Metro Ethernet Forum (MEF) standards group.

The router supports the following:

- Delay Measurement (DM)
- Synthetic Loss Measurement (SLM)

Two-Way Delay Measurement for Scalability

Use the Ethernet frame delay measurement to measure frame delay and frame delay variations. The system measures the Ethernet frame delay by using the Delay Measurement Message (DMM) method.

Restrictions for Configuring Two-Way Delay Measurement

Follow the guidelines and restrictions listed here when you configure two-way delay measurement:

- Y.1731 PM does not support One-Way DMM since PTP support is not available in the Release 6.3.1 for NCS 5500.

- System supports software-based timestamping for Two-Way DMM for NCS5502 and NCS5508 routers. The restriction is only applicable to UP MEP (Maintenance association End Point), which requires core NPU (Network Processor) and access NPU to have ToD (Time of Day) in sync to support 64-bit hardware-based timestamping. After you enable PTP (Precision Time Protocol) and sync all NPUs, the restriction is removed.

Configuring Two-Way Delay Measurement

Perform the following steps to configure two-way delay measurement:

RP/0/RP0/CPU0:router **(config)** # **ethernet sla**

```
profile DMM type cfm-delay-measurement
  probe
   send burst every 5 seconds packet count 5 interval 1 seconds
  !
  schedule
   every 1 minutes for 40 seconds
  !
  statistics
   measure round-trip-delay
    buckets size 1 probes
    buckets archive 5
   !
   measure round-trip-jitter
    buckets size 1 probes
    buckets archive 1
   !
!
!
!
interface TenGigE0/0/0/10.1 l2transport
encapsulation dot1q 1
ethernet cfm
  mep domain DOWN0 service s10 mep-id 2001
   sla operation profile DMM target mep-id 6001
  !
```

Configuring an On-Demand Ethernet SLA Operation for CFM Delay Measurement

To configure an on-demand Ethernet SLA operation for CFM delay measurement, use this command in privileged EXEC configuration mode:

RP/0/RP0/CPU0:router (config) #

```
ethernet sla on-demand operation type cfm-synthetic-loss-measurement probe domain D1 source
 interface TenGigE 0/6/1/0 target mac-address 2.3.4
```

Running Configuration

```
RP/0/RP0/CPU0:router# show ethernet cfm peer meps
Mon Sep 11 12:09:44.534 UTC
Flags:
> - Ok                              I - Wrong interval
R - Remote Defect received          V - Wrong level
L - Loop (our MAC received)         T - Timed out
C - Config (our ID received)        M - Missing (cross-check)
X - Cross-connect (wrong MAID)      U - Unexpected (cross-check)
* - Multiple errors received        S - Standby

Domain UP6 (level 6), Service s6
```

```
Up MEP on FortyGigE0/0/1/2.1 MEP-ID 1
================================================================================
St    ID MAC Address    Port    Up/Downtime   CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ----------- --------- ------ ----- -----
 >  4001 70e4.227c.2865 Up      00:01:27            0      0     0     0

Domain DOWN0 (level 0), Service s10
Down MEP on TenGigE0/0/0/10.1 MEP-ID 2001
================================================================================
St    ID MAC Address    Port    Up/Downtime   CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ----------- --------- ------ ----- -----
 >  6001 70e4.227c.287a Up      00:02:11            0      0     0     0
RP/0/RP0/CPU0:router#
RP/0/RP0/CPU0:router# show running-config
Mon Sep 11 12:10:18.467 UTC
interface FortyGigE0/0/1/2.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain UP6 service s6 mep-id 1
    sla operation profile DMM target mep-id 6001
    sla operation profile test-slm target mep-id 6001
  !
 !
!
l2vpn
 xconnect group g1
  p2p p1
   interface TenGigE0/0/0/10.1
   interface FortyGigE0/0/1/2.1
  !
 !
!
end
```

Verification

```
One-way Delay (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 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: 1912765.961ms; Max: 1912765.961ms; Mean: 1912765.961ms; StdDev: -2147483.648ms


One-way Delay (Dest->Source)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 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: -1912765.952ms; Max: -1912765.951ms; Mean: -1912765.951ms; StdDev: -2147483.648ms


Round Trip Jitter
~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
```

```
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 9
    Min: 0.000ms; Max: 0.001ms; Mean: 0.000ms; StdDev: 0.000ms


One-way Jitter (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 9
    Min: 0.000ms; Max: 0.000ms; Mean: 0.000ms; StdDev: 0.000ms


One-way Jitter (Dest->Source)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 9
    Min: 0.000ms; Max: 0.001ms; Mean: 0.000ms; StdDev: 0.000ms


RP/0/RP0/CPU0:ios#ethernet sla on-demand operation type cfm-syn probe domain DOWN0 source
interface tenGigE 0/0/0/10.1 target mep-id 6001
Mon Sep 11 12:12:39.259 UTC
Warning: Burst configuration is present and so this profile cannot be represented in the
MEF-SOAM-PM-MIB configuration tables. However, the statistics are still collected
On-demand operation 2 succesfully created
 / - Completed - statistics will be displayed shortly.
RP/0/RP0/CPU0:ios#show ethernet sla statistics on-demand id 2

Mon Sep 11 12:13:24.825 UTC
Source: Interface TenGigE0/0/0/10.1, Domain DOWN0
Destination: Target MEP-ID 6001
================================================================================
On-demand operation ID #2, packet type 'cfm-synthetic-loss-measurement'
Started at 12:12:41 UTC Mon 11 September 2017, runs once for 10s
Frame Loss Ratio calculated every 10s

One-way Frame Loss (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:12:41 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 100; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 1
    Min: 0.000%; Max: 0.000%; Mean; 0.000%; StdDev: 0.000%; Overall: 0.000%


One-way Frame Loss (Dest->Source)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:12:41 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 100; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 1
    Min: 0.000%; Max: 0.000%; Mean; 0.000%; StdDev: 0.000%; Overall: 0.000%
```

```
RP/0/RP0/CPU0:ios#show ethernet cfm local meps verbose
Mon Sep 11 12:13:04.461 UTC
Domain UP6 (level 6), Service s6
Up MEP on FortyGigE0/0/1/2.1 MEP-ID 1
================================================================================
  Interface state: Up      MAC address: 008a.960f.c4a8
  Peer MEPs: 1 up, 0 with errors, 0 timed out (archived)
  Cross-check errors: 0 missing, 0 unexpected

  CCM generation enabled:  Yes, 1s (Remote Defect detected: No)
                           CCM processing offloaded to hardware
  AIS generation enabled:  No
  Sending AIS:             No
  Receiving AIS:           No

  No packets sent/received

Domain DOWN0 (level 0), Service s10
Down MEP on TenGigE0/0/0/10.1 MEP-ID 2001
================================================================================
  Interface state: Up      MAC address: 008a.960f.c428
  Peer MEPs: 1 up, 0 with errors, 0 timed out (archived)
  Cross-check errors: 0 missing, 0 unexpected

  CCM generation enabled:  Yes, 1s (Remote Defect detected: No)
                           CCM processing offloaded to hardware
  AIS generation enabled:  No
  Sending AIS:             No
  Receiving AIS:           No

  Packet        Sent      Received
  ------  ----------  ---------------------------------------------------------
  DMM             10             0
  DMR              0            10
  SLM            100             0
  SLR              0           100
```

Synthetic Loss Measurement

The synthetic loss measurement mechanism defined in Y.1731 can only be used in point-to-point networks, and only works when there is sufficient flow of data traffic. The difficulties with the Y.1731 loss measurement mechanism was recognized across the industry and hence an alternative mechanism has been defined and standardized for measuring loss of traffic.

This alternative mechanism does not measure the loss of the actual data traffic, but instead injects synthetic CFM frames and measures the loss of these synthetic frames. You can perform a statistical analysis to give an approximation of the loss of data traffic. This technique is called Synthetic Loss Measurement (SLM). SLM has been included in the latest version of the Y.1731 standard. Use SLA to perform the following measurements:

- One-way loss (Source to Destination)
- One-way loss (Destination to Source)

SLM supports the following:

- All Layer 2 transport interfaces, such as physical, bundle interfaces, Layer2 sub-interfaces, pseudowire Head-end interfaces or attachment circuits.

- Up and Down MEPs.
- Transparent passing of the SLM packets through the MIP without punting it to the software.
- 1000 pps of SLM/SLR traffic.

Configuring Synthetic Loss Measurement

The following section describes how you can configure Synthetic Loss Measurement:

RP/0/RP0/CPU0:router **(config)# ethernet sla**

```
profile test-slm type cfm-synthetic-loss-measurement
  probe
   send packet every 1 seconds
   synthetic loss calculation packets 24
  !
  schedule
   every 3 minutes for 120 seconds
  !
  statistics
   measure one-way-loss-sd
    buckets size 1 probes
    buckets archive 5
   !
   measure one-way-loss-ds
    buckets size 1 probes
    buckets archive 5
!
!
!
!
interface TenGigE0/0/0/10.1 l2transport
encapsulation dot1q 1
ethernet cfm
  mep domain DOWN0 service s10 mep-id 2001
   sla operation profile test-slm target mep-id 6001
  !
```

Configuring an On-Demand Ethernet SLA Operation for CFM Synthetic Loss Measurement

To configure an on-demand Ethernet SLA operation for CFM synthetic loss measurement, use this command in privileged EXEC configuration mode:

RP/0/RP0/CPU0:router (config) # **ethernet sla on-demand operation type cfm-synthetic-loss-measurement probe domain D1 source interface TenGigE 0/6/1/0 target mac-address 2.3.4**

Running Configuration

```
RP/0/RP0/CPU0:router# show ethernet sla statistics on-demand id 1
Mon Sep 11 12:12:00.699 UTC
Source: Interface TenGigE0/0/0/10.1, Domain DOWN0
Destination: Target MEP-ID 6001
================================================================================
On-demand operation ID #1, packet type 'cfm-delay-measurement'
RP/0/RP0/CPU0:router#
RP/0/RP0/CPU0:router# show running-config
Mon Sep 11 12:10:18.467 UTC
Building configuration...
```

```
!! IOS XR Configuration version = 6.4.1.14I
!! Last configuration change at Mon Sep 11 12:08:16 2017 by root
!
logging console disable
telnet vrf default ipv4 server max-servers 10
username root
group root-lr
group cisco-support
secret 5 $1$QJT3$94M5/wK5J0v/lpAu/wz31/
!
line console
exec-timeout 0 0
!
ethernet cfm
domain UP6 level 6 id null
  service s6 xconnect group g1 p2p p1 id number 6
   mip auto-create all ccm-learning
   continuity-check interval 1s
   mep crosscheck
    mep-id 4001
   !
  !
!
domain DOWN0 level 0 id null
 service s10 down-meps id number 10
   continuity-check interval 1s
   mep crosscheck
    mep-id 6001
   !
  !
!
!
profile test-slm type cfm-synthetic-loss-measurement
  probe
   send packet every 1 seconds
   synthetic loss calculation packets 24
  !
  schedule
   every 3 minutes for 120 seconds
  !
  statistics
   measure one-way-loss-sd
    buckets size 1 probes
    buckets archive 5
   !
   measure one-way-loss-ds
    buckets size 1 probes
    buckets archive 5
!
interface MgmtEth0/RP0/CPU0/0
shutdown
!
interface TenGigE0/0/0/0
shutdown
!
interface TenGigE0/0/0/1
shutdown
!
interface TenGigE0/0/0/2
shutdown
!
interface TenGigE0/0/0/3
shutdown
!
```

```
interface TenGigE0/0/0/4
shutdown
!
interface TenGigE0/0/0/5
shutdown
!
interface TenGigE0/0/0/6
shutdown
!
interface TenGigE0/0/0/7
shutdown
!
interface TenGigE0/0/0/8
shutdown
!
interface TenGigE0/0/0/9
shutdown
!
interface TenGigE0/0/0/10.1 l2transport
encapsulation dot1q 1
ethernet cfm
  mep domain DOWN0 service s10 mep-id 2001
   sla operation profile DMM target mep-id 6001
   sla operation profile test-slm target mep-id 6001
  !
!
!
interface TenGigE0/0/0/11
shutdown
!
interface TenGigE0/0/0/12
shutdown
!
interface TenGigE0/0/0/13
shutdown
!
interface TenGigE0/0/0/14
shutdown
!
interface TenGigE0/0/0/15
shutdown
!
interface TenGigE0/0/0/16
shutdown
!
interface TenGigE0/0/0/17
shutdown
!
interface TenGigE0/0/0/18
shutdown
!
interface TenGigE0/0/0/19
shutdown
!
interface TenGigE0/0/0/20
shutdown
!
interface TenGigE0/0/0/21
shutdown
!
interface TenGigE0/0/0/22
shutdown
!
interface TenGigE0/0/0/23
```

```
shutdown
!
interface TenGigE0/0/0/24
shutdown
!
interface TenGigE0/0/0/25
shutdown
!
interface TenGigE0/0/0/26
shutdown
!
interface TenGigE0/0/0/27
shutdown
!
interface TenGigE0/0/0/28
shutdown
!
interface TenGigE0/0/0/29
shutdown
!
interface TenGigE0/0/0/30
shutdown
!
!
interface TenGigE0/0/0/31
 shutdown
!
interface TenGigE0/0/0/32
 shutdown
!
interface TenGigE0/0/0/33
 shutdown
!
interface TenGigE0/0/0/34
 shutdown
!
interface TenGigE0/0/0/35
 shutdown
!
interface TenGigE0/0/0/36
 shutdown
!
interface TenGigE0/0/0/37
 shutdown
!
interface TenGigE0/0/0/38
 shutdown
!
interface TenGigE0/0/0/39
 shutdown
!
interface TenGigE0/0/1/0/1
 shutdown
!
interface TenGigE0/0/1/0/2
 shutdown
!
interface TenGigE0/0/1/0/3
 shutdown
!
controller Optics0/0/1/0
 breakout 4x10
!
interface HundredGigE0/0/1/1
```

```
 shutdown
!
interface FortyGigE0/0/1/2.1 l2transport
 encapsulation dot1q 1
 ethernet cfm
  mep domain UP6 service s6 mep-id 1
    sla operation profile DMM target mep-id 6001
    sla operation profile test-slm target mep-id 6001
  !
 !
!
l2vpn
 xconnect group g1
  p2p p1
   interface TenGigE0/0/0/10.1
   interface FortyGigE0/0/1/2.1
  !
 !
!
end
```

Verification

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

Bucket started at 12:11:19 UTC Mon 11 September 2017 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: 0.009ms; Max: 0.010ms; Mean: 0.009ms; StdDev: 0.000ms


One-way Delay (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 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: 1912765.961ms; Max: 1912765.961ms; Mean: 1912765.961ms; StdDev: -2147483.648ms


One-way Delay (Dest->Source)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 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: -1912765.952ms; Max: -1912765.951ms; Mean: -1912765.951ms; StdDev: -2147483.648ms


Round Trip Jitter
~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 9
```

```
    Min: 0.000ms; Max: 0.001ms; Mean: 0.000ms; StdDev: 0.000ms


One-way Jitter (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 9
    Min: 0.000ms; Max: 0.000ms; Mean: 0.000ms; StdDev: 0.000ms


One-way Jitter (Dest->Source)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket

Bucket started at 12:11:19 UTC Mon 11 September 2017 lasting 10s
    Pkts sent: 10; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 9
    Min: 0.000ms; Max: 0.001ms; Mean: 0.000ms; StdDev: 0.000ms
```

CFM and Y 1731 on VPLS over BGP Signaling

Table 6: Feature History Table

Feature Name	Release Information	Feature Description
CFM and Y 1731 on VPLS over BGP Signaling	Release 7.6.1	VPLS over BGP Signaling services supports CFM continuity check, ITU-T Y.1731 compliant Delay Measurement Message (DMM), and Synthetic Loss Measurement (SLM) functions. This feature allows you to effectively manage a network with L2VPN services running VPLS using BGP AD.

Connectivity fault management (CFM) is a service-level Operations and Maintenance (OAM) protocol that provides tools for monitoring and troubleshooting end-to-end Ethernet services. This feature provides high-speed Layer 2 and Layer 3 services with high resiliency and less operational complexity to different market segments.

The CFM on VPLS over BGP Signaling feature allows you to effectively manage a network with L2VPN services running VPLS. The CFM provides proactive network management, troubleshooting, connectivity monitoring, fault verification, and fault isolation.

CFM on VPLS services supports CFM continuity check, ITU-T Y.1731 compliant Delay Measurement Message (DMM), and Synthetic Loss Measurement (SLM) functions.

DMM is used to periodically measure frame delay and frame delay variation between a pair of point-to-point Maintenance End Point (MEPs). Measurements are made between two MEPs belonging to the same domain and Maintenance Association (MA).

SLM is used to periodically measure Frame Loss and Forward Loss Ratio (FLR) between a pair of point-to-point MEPs. Measurements are made between two MEPs that belong to the same domain and MA.

Supported Offload Types and Timer Values

The following are supported offload types:

- Hardware (HW) Offload type: The check message (CCM) timers for a CFM session are 3.3ms 10ms, 100ms, or 1s.
- Non-Offload type: The CCM timers for a CFM session on a physical interface are equal to 10s or 1m.
- Software (SW) Offload type: The CFM session on a bundle interface. SW Offload type supports 1s, 10s, or 1m.

The following are the supported timer values:

- 3.3ms: Interval of 3.3 milliseconds
- 10ms: Interval of 10 milliseconds
- 100ms: Interval of 100 milliseconds
- 1s: Interval of 1 second
- 10s: Interval of 10 seconds
- 1m: Interval of 1 minute

Feature Highlights

- CFM and Y 1731 on VPLS over BGP Signaling is now supported only on routers that have Cisco NC57 line cards that are installed and operate in native mode only.
- Supports single homing with one AC per PW.
- Support 1 and 2 Way DMM and SLM for UP and Down MEPs

Restrictions

- Supports single homing with one AC per PW.
- Supports 1 Way DMM for the hardware with support for timing sync.
- The Cisco NC57 line cards operating in native mode support hardware timestamping only when the RP card is used as an RP-E card. With non-RP-E cards, the Cisco NC 57 line cards perform software timestamping and Delay Measurement Message (DMM) results have higher value for Mean, Maximum, and Minimum.

Configure CFM and Y 1731 on VPLS over BGP Signaling

Configuration Example

```
/* BGP AD based VPLS with single AC.
 */
l2vpn
 bridge group cfmvpls
  bridge-domain cfmvpls1
```

```
   interface Bundle-Ether203.6001
   !
   vfi cfmvpls1
    vpn-id 1001
    autodiscovery bgp
     rd auto
     route-target 1001:1001
     signaling-protocol bgp
      ve-id 1


/* Global CFM UP MEP configuration */
ethernet cfm
 domain cfmvpls level 3 id null
  service cfmvpls1 bridge group cfmvpls bridge-domain cfmvpls1 id number 50001
   continuity-check interval 1s loss-threshold 3
   mep crosscheck
    mep-id 4000


/* Global CFM DOWN MEP configuration */
ethernet cfm
 domain cfmvplsdown level 3 id null
  service cfmvplsdown1 down-meps id number 29001
   continuity-check interval 1s loss-threshold 3
   mep crosscheck
    mep-id 4000


/* Global Y1731 DMM Configuration */
ethernet sla
 profile dmm1 type cfm-delay-measurement
  probe
   send burst every 1 minutes packet count 30 interval 2 seconds
   priority 4
  schedule
   every 5 minutes for 300 seconds
  statistics
   measure round-trip-delay
   measure one-way-delay-sd
   !
   measure one-way-delay-ds
   measure round-trip-jitter
   measure one-way-jitter-sd
   measure one-way-jitter-ds


/* Global Y1731 SLM Configuration */
ethernet sla
 profile eth_sla_slm type cfm-synthetic-loss-measurement
  probe
   send burst every 1 minutes packet count 60 interval 1 seconds
   priority 7
  !
  schedule
   every 5 minutes for 300 seconds
  !
  statistics
   measure one-way-loss-sd
   !
   measure one-way-loss-ds
   !



/* CFM UP MEP or DOWN MEP and Ethernet SLA applied to interface  */
```

```
interface Bundle-Ether203.6001 l2transport
 encapsulation dot1q 4002 second-dot1q 1
 rewrite ingress tag pop 2 symmetric
 ethernet cfm
  mep domain cfmvpls service cfmvpls1 mep-id 1
    sla operation profile dmm1 target mep-id 4000
  !
  mep domain cfmvplsdown service cfmvplsdown1 mep-id 1
    sla operation profile eth_sla_slm target mep-id 4000
```

Verification Example

Example output with the CFM Up MEP is configured.

```
Router(PE1)# show ethernet cfm peer meps interface bundle-Ether 203.6001
Flags:
> - Ok                          I - Wrong interval
R - Remote Defect received      V - Wrong level
L - Loop (our MAC received)     T - Timed out
C - Config (our ID received)    M - Missing (cross-check)
X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
* - Multiple errors received    S - Standby
Domain cfmvpls (level 3), Service cfmvpls1
Up MEP on Bundle-Ether203.6001 MEP-ID 1
================================================================================
St    ID MAC Address    Port    Up/Downtime   CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ----------- --------- ------ ----- -----
 >  4000 d46d.5059.1db0 Up      15:33:42        56055      0    0     0
```

Example output with the CFM Down MEP is configured.

```
Router(PE1)#show ethernet cfm peer meps interface bundle-Ether 203.6001
Flags:
> - Ok                          I - Wrong interval
R - Remote Defect received      V - Wrong level
L - Loop (our MAC received)     T - Timed out
C - Config (our ID received)    M - Missing (cross-check)
X - Cross-connect (wrong MAID)  U - Unexpected (cross-check)
* - Multiple errors received    S - Standby

Domain cfmvplsdown (level 3), Service cfmvplsdown1
Down MEP on Bundle-Ether203.6001 MEP-ID 1
================================================================================
St    ID MAC Address    Port    Up/Downtime   CcmRcvd SeqErr   RDI Error
-- ----- -------------- ------- ----------- --------- ------ ----- -----
 >  4000 0024.f71d.af3e Up      15:37:33       112487      0     0     0
```

Example output with the Ethernet SLA DMM statistics.

```
Router(PE1)#show ethernet sla statistics interface bundle-Ether 203.6001 domain cfmvpls
profile  dmm1
Source: Interface Bundle-Ether203.6001, Domain cfmvpls
Destination: Target MEP-ID 4000
===========================================================================
Profile 'dmm1', packet type 'cfm-delay-measurement'
Scheduled to run every 5min first at 00:03:31 UTC for 5min
Round Trip Delay
~~~~~~~~~~~~~~~~
1 probes per bucket
No stateful thresholds.
Bucket started at 03:18:31 IST Mon 14 February 2022 lasting 5min
    Pkts sent: 150; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
```

```
    Result count: 150
    Min: 290857.011ms; Max: 291925.308ms; Mean: 291367.479ms; StdDev: 317.339ms
```

Example output with the Ethernet SLA SLM statistics.

```
Router(PE1)#show ethernet sla statistics interface bundle-Ether 203.6001 domain cfmvplsdown
 profile eth_sla_slm
Source: Interface Bundle-Ether203.6001, Domain cfmvplsdown
Destination: Target MEP-ID 4000
================================================================================
Profile 'eth_sla_slm', packet type 'cfm-synthetic-loss-measurement'
Scheduled to run every 5min first at 00:01:50 UTC for 5min
Frame Loss Ratio calculated every 5min
One-way Frame Loss (Source->Dest)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 probes per bucket
No stateful thresholds.
Bucket started at 03:21:50 IST Mon 14 February 2022 lasting 5min
    Pkts sent: 300; Lost: 0 (0.0%); Corrupt: 0 (0.0%);
                   Misordered: 0 (0.0%); Duplicates: 0 (0.0%)
    Result count: 1
    Min: 0.000%; Max: 0.000%; Mean; 0.000%; StdDev: 0.000%; Overall: 0.000%
```

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 protocol to detect link problems, which are not detected at the physical link layer. This protocol is specifically targeted at possible wiring errors, when using unbundled fiber links, where there can be a mismatch between the transmitting and receiving connections of a port.

Limitations

- UDLD must not be enabled on a Switched Port Analyzer (SPAN) source or a destination port.
- UDLD must not be enabled on a port that acts as a source or destination port for SPAN.

Types of Fault Detection

UDLD can detect these types of faults:

- Transmit faults — These are cases where there is a failure in transmitting packets from the local port to the peer device, but packets are being received from the peer. These faults are caused by failure of the physical link (where notification at layer 1 of unidirectional link faults is not supported by the media) as well as packet path faults on the local or peer device.
- Miswiring faults — These are cases where 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). This can occur when using unbundled fibers to connect fiber optic ports.
- Loopback faults — These are cases where 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 includes 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 which determine the behavior on a heartbeat timeout. These modes are described in the section UDLD Modes of Operation, on page 97.

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.

Note The difference of behavior between normal and aggressive modes is only seen in case of neighbor timeout. In all other cases, irrespective of the normal or aggressive mode, the system error disables a link once a unidirectional link is detected.

Configure UDLD

UDLD is configured for each interface. The interface must be a physical ethernet interface.

Perform the following steps to configure UDLD protocol on an interface:

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

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

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

Running Configuration

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

RP/0/RP0/CPU0:IOS(config)#interface tenGigE 0/0/0/0

 RP/0/RP0/CPU0:IOS(config-if)#ethernet udld

RP/0/RP0/CPU0:IOS(config-if-udld)#mode ?

   aggressive  Run UDLD in aggressive mode

   normal      Run UDLD in normal mode

 RP/0/RP0/CPU0:IOS(config-if-udld)#mode aggressive
 RP/0/RP0/CPU0:IOS(config-if-udld)#message-time ?

   <7-90>  'Mslow' message time (in seconds) to use for the UDLD protocol

 RP/0/RP0/CPU0:IOS(config-if-udld)#message-time 50

RP/0/RP0/CPU0:IOS(config-if-udld)#destination mac-address ?
   H.H.H
             A valid multicast MAC address

   cisco-l2cp           Use the Cisco L2CP MAC address (used by CDP)

   ieee-slow-protocols  Use the IEEE slow protocol destination MAC address
RP/0/RP0/CPU0:IOS(config-if-udld)#destination mac-address 0100.5e01.0101

 RP/0/RP0/CPU0:IOS(config-if-udld)#logging disable

 RP/0/RP0/CPU0:IOS(config-if-udld)#commit

 RP/0/RP0/CPU0:IOS(config-if-udld)#end

 RP/0/RP0/CPU0:IOS#sh run interface tenGigE 0/0/0/0
 interface TenGigE0/0/0/0

  ethernet udld

  mode aggressive

   message-time 50

  destination mac-address 0100.5e01.0101

   logging disable

  !

!
```

Verification

```
RP/0/RP0/CPU0:IOS#sh ethernet udld interfaces

 Device ID:               00:8a:96:e1:20:d8

 Device name:             IOS

 Interface TenGigE0/0/0/0

Port state:            Up

Main FSM state:        Advertising

Detection FSM state:   Unknown

Message interval:      7 seconds

Timeout interval:      5 seconds

Destination MAC:       01:00:5e:01:01:01


RP/0/RP0/CPU0:IOS#sh ethernet udld statistics tenGigE 0/0/0/0
```

```
Interface TenGigE0/0/0/0

Counters last cleared:               00:01:18 ago

Main FSM transitions (to each state)

Link up:     1

Detection:   0

Advertise:   1

Port shutdown: 0

UDLD inactive: 0

Detection FSM transitions (to each state)

Unknown: 0

Bidirectional: 0

Unidirectional: 0

Neighbor mismatch: 0

Loopback: 0

Rx packet counts

Probe: 0

Echo:                                         0

Flush:                                        0

Invalid packets (dropped):                    0

Tx packet counts

Probe:                                       19

Echo:                                         0

Flush:                                        0

Unable to send (dropped):                     0

RP/0/RP0/CPU0:IOS#

 RP/0/RP0/CPU0:IOS#sh ethernet udld daemon database

Interface TenGigE0/0/0/0


   Item                          Value

----------------------------  -----------------------------------------------

  Interface handle              Te0/0/0/0 (0x00000200)
  Name                          Te0/0/0/0
  Name (long internal format)   TenGigE0_0_0_0
  Configured ?                  TRUE
  Caps add in progress ?        FALSE
  Caps remove in progress ?     FALSE
  Caps added ?                  TRUE
  Protocol start pending ?      FALSE
  Protocol running ?            TRUE
  Registered for packet I/O ?   TRUE
  Aggressive mode ?             TRUE
  Logging enabled ?             FALSE
  Error disabled on start ?     FALSE
  Error disabled during ISSU ?  FALSE
  Attributes read ?             TRUE
  Pending state down nfn ?      FALSE
  Message time                  50
```

Ethernet SLA Statistics Measurement in a Profile

Table 7: 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 that are used to aggregate the results of Ethernet SLA statistics, in microseconds. The size of the bins is defined by the **width** value of delay and jitter measurement in Ethernet SLA statistics. You can configure the **width** value ranging from 1 to 10000000 microseconds. This enhancement provides granularity to store more accurate results of Ethernet 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%)        -
```

Link Loss Forwarding

The Cisco ASR9000 Series Routers support Link Loss Forwarding (LLF). LLF uses CFM to transmit notification of a signal loss or fault across the network. If a local AC goes down, LLF sends signals across to the neighboring device.

The following packet types indicate a fault in a network:

- Continuity Check Message (CCM). Also known as CFM
- Alarm Indication Signal (AIS)
- Client Signal Frame (CSF)

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

Ether-MA handles owner channel communication and resyncs from CFMD, L2VPN, and other Ether- MA processes.

Restrictions for LLF

- LLF isn't permitted on sub-interfaces.
- LLF is only permitted on up MEPs.
- LLF is not supported on LACP bundle.

- The system runs a damping timer to govern transitions from an interface being TX-disabled to an interface being TX-enabled. The following restrictions apply on such a scenario:
  - The period of the damping timer is given by three times the configured CCM interval. However, you can't configure the damping timer.
  - The system doesn't provide damping for transitions from TX-enabled to TX-disabled.

Configure Link Loss Forwarding

This section describes how to configure LLF on a network by using the `propagate-remote-status config` command.

```
/* Enable LLF */
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet0/2/0/0
RP/0/RP0/CPU0:router(config-cfm)# ethernet cfm
RP/0/RP0/CPU0:router(config-cfm)# mep domain foo service bar mep-id 1
RP/0/RP0/CPU0:router(config-cfm)# propagate remote-status
RP/0/RP0/CPU0:router(config-cfm)# commit
```

Optional Configuration for Client Signal Fail (CSF)

Note CSF configuration is required for inter-operation with certain client-end setups that contain devices from other clients.

```
ethernet cfm
  domain <domain> level <level> service <service> <type>
    csf [<interval> {1s | 1m}] [cos <cos>]
      log csf
```

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

```
show ethernet cfm interfaces [ <interface> ] llf [ location <node> ]
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
```