Configuring Ethernet OAM on the Cisco ASR 9000 Series Router

This module describes the configuration of Ethernet Operations, Administration, and Maintenance (OAM) on the Cisco ASR 9000 Series Aggregation Services Routers.

Feature History for Configuring Ethernet OAM

<table>
<thead>
<tr>
<th>Release</th>
<th>Modification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release 3.7.2</td>
<td>Support for the following features was introduced:</td>
</tr>
<tr>
<td></td>
<td>• Ethernet Link OAM</td>
</tr>
<tr>
<td></td>
<td>• Ethernet CFM</td>
</tr>
<tr>
<td>Release 3.7.3</td>
<td>Support for the CFM Exploratory Linktrace feature was introduced.</td>
</tr>
<tr>
<td>Release 3.9.0</td>
<td>Support for the Ethernet SLA feature was introduced.</td>
</tr>
<tr>
<td>Release 3.9.1</td>
<td>Support for the following features was introduced:</td>
</tr>
<tr>
<td></td>
<td>• Ethernet CFM on Link Aggregation Group (LAG) interfaces (Ethernet bundle interfaces), Ethernet and bundle subinterfaces, and LAG member (bundle member) interfaces.</td>
</tr>
<tr>
<td></td>
<td>• EFD</td>
</tr>
<tr>
<td></td>
<td>• AIS</td>
</tr>
<tr>
<td></td>
<td>• Flexible tagging</td>
</tr>
<tr>
<td></td>
<td>• The <code>ethernet cfm mep domain</code> command is replaced by the <code>ethernet cfm</code> and <code>mep domain</code> commands.</td>
</tr>
</tbody>
</table>
Release 4.0.0
Support for the following features was introduced:

- The `action link-fault` command is replaced by the `action uni-directional link fault` command.
- The `efd` keyword is added to put an interface into the line protocol down state, as an option for the following commands:
  - `action capabilities-conflict`
  - `action discovery-timeout`
  - `action session-down`
  - `action uni-directional link-fault`
- Uni-directional link-fault detection to identify local link-faults and send notification to a remote Ethernet OAM peer using the `uni-directional link-fault detection` command.
- Support for the following enhancements to Ethernet SLA was added:
  - Support for on-demand Ethernet SLA operations using the `ethernet sla on-demand operation` commands.
  - One-way delay and jitter measurements using the following new keyword options for the `statistics measure` command:
    - `one-way-delay-ds`
    - `one-way-delay-sd`
    - `one-way-jitter-ds`
    - `one-way-jitter-sd`
  - Specification of a test pattern to pad loopback packets when measuring delay.
  - Displaying the time when the minimum (Min) and maximum (Max) values of a statistic occurred in the measurement time period in the `show ethernet sla statistics detail` command.

Release 4.0.1
Support for Ethernet CFM on Multi-Chassis Link Aggregation Groups (MC-LAG) was added.

Release 4.1.0
Support for the following feature was introduced:

- E-LMI
- Timestamps for delay packets were changed from being derived by the system time-of-day (NTP) clock to the DTI timing input on the clock-interfaces on the RSP.

Release 4.2.0
Support for Unidirectional Link Detection Protocol (UDLD) was added.

Release 4.2.2
Support for Unidirectional Link Routing (UDLR) was included.
Prerequisites for Configuring Ethernet OAM

You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

Before configuring Ethernet OAM, confirm that at least one of the Gigabit Ethernet line cards supported on the router is installed:

- 2-Port 10-Gigabit Ethernet, 20-Port Gigabit Ethernet Combination line card (A9K-2T20GE-B and A9K-2T20GE-L)
- 4-Port 10-Gigabit Ethernet line card (A9K-4T-L, -B, or -E)
- 8-Port 10-Gigabit Ethernet DX line card (A9K-8T/4-L, -B, or -E)
- 8-Port 10-Gigabit Ethernet line card (A9K-8T-L, -B, or -E)
- 16-Port 10-Gigabit Ethernet SFP+ line card (A9K-16T/8-B and A9K-16T/8-B+AIP)
- 40-Port Gigabit Ethernet line card (A9K-40GE-L, -B, or -E)
Information About Configuring Ethernet OAM

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

- Ethernet Link OAM, page 68
- Ethernet CFM, page 70
- Ethernet SLA (Y.1731 Performance Monitoring), page 89
- Ethernet LMI, page 93
- Unidirectional Link Detection Protocol, page 97
- Unidirectional Link Routing, page 99

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, take actions on events, and if necessary, put specific interfaces into loopback mode for troubleshooting. 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 EOAM 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.

The following standard Ethernet Link OAM features are supported on the router:

- Neighbor Discovery, page 69
- Link Monitoring, page 69
- MIB Retrieval, page 69
- Miswiring Detection (Cisco-Proprietary), page 69
- Remote Loopback, page 69
- SNMP Traps, page 69
- Unidirectional Link Fault Detection, page 69
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.

Link Monitoring

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

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.

Remote Loopback

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

SNMP Traps

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

Unidirectional Link Fault Detection

Unidirectional link fault detection describes an Ethernet link OAM function that runs directly on physical Ethernet interfaces (not VLAN subinterfaces or bundles) that uses a defined link fault message to signal link faults to a remote host. Unidirectional link fault detection offers similar functionality to Gigabit Ethernet hardware-level signaling of a link fault, but it is done at a higher protocol layer as part of Ethernet link OAM. The hardware function uses the Remote Fault Indication bit set in a frame that is signaled out-of-band, where unidirectional link fault detection signals the error using an OAMPDU.

Unidirectional link fault detection only applies to a single, physical link. When the remote host receives the link fault message, the interface can be shut down for all higher-layer protocols, and specifically, Layer 2 switching and Layer 3 routing protocols. While the fault is detected, a link fault message is sent periodically to the remote host. Once a fault is no longer detected, the link fault message is no longer sent, and the remote host can bring back the interface.
Unidirectional link fault detection is configured using the **uni-directional link-fault detection** command, and does not affect how the receipt of link-fault messages are handled by the router. Actions to be taken for the receipt of link-fault messages are configured using the **action uni-directional link-fault** command.

### Ethernet CFM

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

CFM is defined in two standards:
- **IEEE 802.1ag**—Defines the core features of the CFM protocol.
- **ITU-T Y.1731**—Redefines, but maintains compatibility with the features of IEEE 802.1ag, and defines some additional features.

Ethernet CFM on the Cisco ASR 9000 Series Router supports the following functions of ITU-T Y.1731:
- **ETH-CC, ETH-RDI, ETH-LB, ETH-LT**—These are equivalent to the corresponding features defined in IEEE 802.1ag.
- **ETH-AIS**—The reception of ETH-LCK messages is also supported.
- **ETH-DM**—This is supported with the Ethernet SLA feature. For more information about Ethernet SLA, see the “Ethernet SLA (Y.1731 Performance Monitoring)” section on page 89.

To understand how the CFM maintenance model works, you need to understand the following concepts and features:
- **Maintenance Domains**, page 71
- **Services**, page 73
- **Maintenance Points**, page 73
- **CFM Protocol Messages**, page 76
- **MEP Cross-Check**, page 83
- **Configurable Logging**, page 84
- **EFD**, page 84
- **Flexible VLAN Tagging for CFM**, page 85
- **CFM on MC-LAG**, page 86

---

**Note**: The Linktrace responder procedures defined in IEEE 802.1ag are used rather than the procedures defined in Y.1731; however, these are interoperable.
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 Figure 1.

*Figure 1*   
*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.
Figure 2 shows an example of the different levels of maintenance domains in a network.

In CFM diagrams, the conventions are that triangles represent MEPs, pointing in the direction that the MEP sends CFM frames, and circles represent MIPs. For more information about MEPs and MIPs, see the “Maintenance Points” section on page 73.

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. Figure 3 illustrates the supported structure for touching and nested domains, and the unsupported intersection of domains.

**Figure 3  Supported CFM Maintenance Domain Structure**

![Diagram showing supported CFM maintenance domain structures](image)

**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 described in the “Maintenance Domains” section on page 71, 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 bridge domain or 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.
Figure 4 illustrates the monitored areas for Down and Up MEPs.

**Figure 4**  
Monitored Areas for Down and Up MEPs

Figure 5 shows maintenance points at different levels. Because domains are allowed to nest but not intersect (see Figure 3), 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.

**Figure 5**  
CFM Maintenance Points at Different Levels

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) or routed (Layer 3) 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), page 76
- Loopback (IEEE 802.1ag and ITU-T Y.1731), page 78
- Linktrace (IEEE 802.1ag and ITU-T Y.1731), page 79
- Exploratory Linktrace (Cisco), page 81
- Alarm Indication Signal (ITU-T Y.1731), page 82
- Delay and Jitter Measurement (ITU-T Y.1731), page 83

**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)” section on page 79.
All the MEPs in a service must transmit CCMs at the same interval. IEEE 802.1ag defines 7 possible intervals that can be used:

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

A MEP detects a loss of connectivity with one of its peer MEPs when some 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.

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

- A configured identifier for the domain of the transmitting MEP. This is referred to as the Maintenance Domain Identifier (MDID).
- A configured identifier for the service of the transmitting MEP. This is referred to as the Short MA Name (SMAN). Together, the MDID and the SMAN make up the Maintenance Association Identifier (MAID). The MAID must be configured identically on every MEP in the service.
- A configured numeric identifier for the MEP (the MEP ID). Each MEP in the service must be configured with a different MEP ID.
- A sequence number.
- A Remote Defect Indication (RDI). Each MEP includes this in the CCMs it is sending, if it has detected a defect relating to the CCMs it is receiving. This notifies all the MEPs in the service that a defect has been detected somewhere in the service.
- The interval at which CCMs are being transmitted.
- The status of the interface where the MEP is operating—for example, whether the interface is up, down, STP blocked, and so on.
Note: The status of the interface (up/down) should not be confused with the direction of any MEPs on the interface (Up MEPs/Down MEPs).

The following defects can be detected from received CCMs:

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

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

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

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

Loopback Messages (LBM) and Loopback Replies (LBR) are used to verify connectivity between a local MEP and a particular remote MP. At the request of the administrator, a local MEP sends unicast LBM messages 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.
Figure 7 shows an example of CFM loopback message flow between a MEP and MIP.

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. Except for one-way delay and jitter measurements, loopback messages can also be used for Ethernet SLA, if the peer does not support delay measurement.

**Note**
The Ethernet CFM loopback function should not be confused with the remote loopback functionality in Ethernet Link OAM (see the “Remote Loopback” section on page 69). CFM loopback is used to test connectivity with a remote MP, and only the CFM LBM packets are reflected back, but Ethernet Link OAM remote loopback is used to test a link by taking it out of normal service and putting it into a mode where it reflects back all packets.

**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.
Figure 8 shows an example of CFM linktrace message flow between MEPs and MIPs.

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.
Exploratory Linktrace (Cisco)

Exploratory Linktrace is a Cisco extension to the standard linktrace mechanism described above. It has two primary purposes:

- Provide a mechanism to locate faults in cases where standard linktrace does not work, such as when a MAC address has never been seen previously in the network. For example, if a new MEP has been provisioned but is not working, standard linktrace does not help isolate a problem because no frames will ever have been received from the new MEP. Exploratory Linktrace overcomes this problem.

- Provide a mechanism to map the complete active network topology from a single node. This can only be done currently by examining the topology (for example, the STP blocking state) on each node in the network individually, and manually combining this information to create the overall active topology map. Exploratory linktrace allows this to be done automatically from a single node.

Exploratory Linktrace is implemented using the Vendor Specific Message (VSM) and Vendor Specific Reply (VSR) frames defined in ITU-T Y.1731. These allow vendor-specific extensions to be implemented without degrading interoperability. Exploratory Linktrace can safely be deployed in a network that includes other CFM implementations because those implementations will simply ignore the Exploratory Linktrace messages.

Exploratory Linktrace is initiated at the request of the administrator, and results in the local MEP sending a multicast Exploratory Linktrace message. Each MP in the network that receives the message sends an Exploratory Linktrace reply. MIPs that receive the message also forward it on. The initiating MEP uses all the replies to create a tree of the overall network topology.

Figure 9 show an example of the Exploratory Linktrace message flow between MEPs.

To avoid overloading the originating MEP with replies in a large network, responding MPs delay sending their replies for a random amount of time, and that time increases as the size of the network increases.
In a large network, there will be a corresponding large number of replies and the resulting topology map will be equally large. If only a part of the network is of interest, for example, because a problem has already been narrowed down to a small area, then the Exploratory Linktrace can be “directed” to start at a particular MP. Replies will thus only be received from MPs beyond that point in the network. The replies are still sent back to the originating MEP.

**Alarm Indication Signal (ITU-T Y.1731)**

Alarm Indication Signal (AIS) messages are used to rapidly notify MEPs when a fault is detected in the middle of a domain, in an event driven way. MEPs thereby learn of the fault much sooner than if they relied on detecting a loss of continuity, for example, failure to receive some number of consecutive CCMs.

Unlike all other CFM messages, AIS messages are injected into the middle of a domain, and sent outward toward the MEPs at the edge of the domain. Typically, AIS messages are injected by a MEP in a lower level domain. To put it another way, when a MEP sends AIS messages, they are sent in the opposite direction to other CFM messages sent by the MEP, and at a level above the MEP’s own level. The AIS messages are received by the MEPs in the higher level domain, not by the peer MEPs in the same domain as the MEP sending the AIS. When a MEP receives an AIS message, it may itself send another AIS message at an even higher level.

Figure 10 show an example of AIS message flow. The maintenance domain levels are numbered at the right side of the diagram.

**Figure 10  AIS Message Flow**

AIS is only applicable in point-to-point networks. In multipoint networks with redundant paths, a failure at a low level does not necessarily result in a failure at a higher level, as the network may reconverge so as to route around the failed link.

AIS messages are typically sent by a MEP. However, AIS messages can also be sent when there is no MEP present, if a fault is detected in the underlying transport, such as if an interface goes down. In ITU-T Y.1731 these are referred to as *server MEPs*.

AIS messages are sent in response to a number of failure conditions:

- Detection of CCM defects, as described “Continuity Check (IEEE 802.1ag and ITU-T Y.1731)” section on page 76.
• Loss of continuity.
• Receipt of AIS messages.
• Failure in the underlying transport, such as when an interface is down.

Received AIS messages can be used to detect and act on failures more quickly than waiting for a loss of continuity. They can also be used to suppress any failure action, on the basis that the failure has already been detected at a lower level and will be handled there. This is described in ITU-T Y.1731; however, the former is often more useful.

**Delay and Jitter Measurement (ITU-T Y.1731)**

The router supports one-way and two-way delay measurement using two packet types:

- Delay Measurement Message (DMM)
- Delay Measurement Response (DMR)

These packets are unicast similar to loopback messages. The packets carry timestamps generated by the system time-of-day clock to support more accurate delay measurement, and also support an SLA manageability front-end. Beginning in Cisco IOS XR Release 4.1, the DDM & DMR packets carry timestamps derived from the DTI timing input on the clock-interface port on the RSP.

However, unlike loopback messages, these message types can also measure one-way delay and jitter either from destination to source, or from source to destination.

For more information about SLA, see the “Ethernet SLA (Y.1731 Performance Monitoring)” section on page 89.

**MEP Cross-Check**

MEP cross-check supports configuration of a set of expected peer MEPs so that errors can be detected when any of the known MEPs are missing, or if any additional peer MEPs are detected that are not in the expected group.

The set of expected MEP IDs in the service is user-defined. Optionally, the corresponding MAC addresses can also be specified. CFM monitors the set of peer MEPs from which CCMs are being received. If no CCMs are ever received from one of the specified expected peer MEPs, or if a loss of continuity is detected, then a cross-check “missing” defect is detected. Similarly, if CCMs are received from a matching MEP ID but with the wrong source MAC address, a cross-check “missing” defect is detected. If CCMs are subsequently received that match the expected MEP ID, and if specified, the expected MAC address, then the defect is cleared.

**Note**

While loss of continuity can be detected for any peer MEP, it is only treated as a defect condition if cross-check is configured.

If cross-check is configured and CCMs are received from a peer MEP with a MEP ID that is not expected, this is detected as a cross-check “unexpected” condition. However, this is not treated as a defect condition.
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.

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 any traffic flowing, 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.
Figure 11 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.

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.
Configuring Ethernet OAM on the Cisco ASR 9000 Series Router

Information About Configuring Ethernet OAM

Figure 12 shows an example of a network with multiple VLANs using CFM.

Figure 12  Service Provider Network With Multiple VLANs and CFM

Figure 12 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).

CFM on MC-LAG

CFM on Multi-Chassis Link Aggregation Groups is supported on the Cisco ASR 9000 Series Router in the following typical network environment:

- The customer edge (CE) device is a dual-homed device that is connected to two provider edge (PE) point-of-attachment (POA) devices. However, the dual-homed device operates without awareness of connectivity to multiple PEs.
- The two points of attachment at the PE form a redundancy group (RG), with one POA functioning as the active POA, and the other as the standby POA for the dual-homed device link.
- As with typical failover scenarios, if a failure occurs with the active POA, the standby POA takes over to retain the dual-homed device’s connectivity to the network.
CFM on MC-LAG support can be qualified at two levels:

- CFM for the RG level—CFM context is per redundancy group and verifies connectivity for the entire RG.
- CFM for the POA level—CFM context is per point of attachment and verifies connectivity to a single POA.

Both levels of CFM support have certain restrictions and configuration guidelines that you must consider for successful implementation.

This section includes the following topics:

- RG-Level CFM, page 87
- POA-Level CFM, page 88
- Supported Features for CFM on MC-LAG, page 88
- Restrictions for CFM on MC-LAG, page 88

For more information about LAG and MC-LAG on the Cisco ASR 9000 Series Router, see the “Configuring Link Bundling on the Cisco ASR 9000 Series Router” chapter in this guide.

**RG-Level CFM**

RG-level CFM is comprised of three areas of monitoring:

- RG Downlink Monitoring, page 87
- RG Uplink Monitoring, page 87
- End-to-End Service Monitoring, page 87

**RG Downlink Monitoring**

RG downlink monitoring uses CFM to verify connectivity between the dual-homed device and the RG. To configure RG downlink monitoring, be sure that the following requirements are met:

- Down MEPs are configured on the bundle.
- Down MEPs on each POA are configured identically, using the same MEP ID and source MAC address.

This configuration has the following restrictions:

- The CCM loss time is greater than the failover time (typically 50 ms), due to the shortest CCM interval of 100 ms that is currently supported, which results in the shortest CCM loss time of 350 ms.

**RG Uplink Monitoring**

RG uplink monitoring uses CFM to verify connectivity from the active POA to the core. To configure RG uplink monitoring, be sure that the following requirements are met:

- Up MEPs are configured on the bundle interface or bundle subinterface on each POA.
- Up MEPs on each POA are configured identically, using the same MEP ID and source MAC address.

**End-to-End Service Monitoring**

End-to-end service monitoring uses CFM to verify the end-to-end service between the dual-homed devices. To configure end-to-end service monitoring, be sure that the following requirements are met:

- A down MEP is configured on the dual-homed device bundle interface or bundle subinterface.
If optional MIPs are configured, then each POA is configured with a MIP on the bundle.
Each POA can have a MIP on the uplink interface (if native Ethernet is used).
The active and standby POA is configured identically.

This configuration has the following restrictions:
• The MIP on the standby POA will not respond to loopback or linktrace requests.

POA-Level CFM

POA-level monitoring uses CFM to verify connectivity between the dual-homed device and a single POA.

To configure POA-level CFM, be sure that the following requirements are met:
• Down MEPs are configured on bundle members only.

This configuration has the following restrictions:
• POA-level monitoring is not supported on uplinks between a single POA and the core.

Supported Features for CFM on MC-LAG

CFM on MC-LAG supports the following CFM features:
• All existing IEEE 802.1ag and Y.1731 functionality on the Cisco ASR 9000 Series Router is supported on an MC-LAG RG.
• CFM maintenance points are supported on an MC-LAG interface. Maintenance points on a standby link are put into standby state.
• Maintenance points in standby state receive CFM messages, but do not send or reply to any CFM messages.
• When a MEP transitions from active to standby, all CCM defects and alarms are cleared.
• Standby MEPs record remote MEP errors and timeouts, but do not report faults. This means that remote MEPs and their errors will appear in show commands, but no logs, alarms, MIB traps, or EFd are triggered and AIS messages are not sent.
• When a MEP transitions from standby to active, any CCM defects previously detected while the MEP was in standby are reapplied and immediate actions are taken (logs, alarms, MIB traps, EFd, and so on).
• CFM on MC-LAG supports the same scale for bundle interfaces that is supported on the Cisco ASR 9000 Series Router.

Restrictions for CFM on MC-LAG

To support CFM on MC-LAG, you must consider the following restrictions and requirements:
• The CFM configuration must be the same on both the active and standby POAs.
• The CFM state is not synchronized between the two POAs. This can lead to flapping of the interface line protocol state on POA failover if EFd is configured. Fault alarms might also be delayed if a failover occurs just after a fault has been detected.
• POA-level CFM monitoring is not supported on a native Ethernet uplink interface.
• MEPs on bundle interfaces at level 0 are not supported.
• Loopback, linktrace, and Y.1731 SLA operations cannot be started from a MEP in standby state.
• Checks for configuration consistency of MEP IDs to ensure identical configuration of POAs is not supported.
• Y.1731 SLA statistics can be split between the two POAs if a failover occurs. An external network management system would need to collect and collate these statistics from the two POAs.

**Ethernet SLA (Y.1731 Performance Monitoring)**

Customers require their service providers to conform to a Service Level Agreement (SLA). Consequently, service providers must be able to monitor the performance characteristics of their networks. Likewise, customers also want to monitor the performance characteristics of their networks. Cisco provides Y.1731 performance monitoring using the Cisco Ethernet SLA feature.

An SLA defines a set of criteria that guarantees a minimum level of service for customers using a service provider network. The criteria can cover many different areas, including latency, jitter, frame loss, and availability.

The Cisco Ethernet SLA feature conforms to the following standards:
• IEEE 802.1ag
• ITU-T Y.1731

The Cisco Ethernet SLA feature provides the architecture to monitor a network at Layer 2. This architecture provides functions such as collecting, storing, displaying, and analyzing SLA statistics. These SLA statistics can be stored and displayed in various ways, so that statistical analysis can be performed.

Ethernet SLA provides the framework for performing the following major functions of performance monitoring:
• Sending probes consisting of one or more packets to measure performance
  Ethernet SLA provides a flexible mechanism for sending SLA probes to measure performance. Probes can consist of either CFM loopback or CFM delay measurement packets. Options are available to modify how often the packets are sent, and to specify the attributes of the probe packets such as the size and priority.
• Scheduling of operations consisting of periodic probes.
  A flexible mechanism is provided by Ethernet SLA to specify how often each probe should be executed, how long it should last, and when the first probe should start. Probes can be scheduled to run back-to-back to provide continuous measurements, or at a defined interval ranging from once a minute to once a week.
• Collecting and storing results.
  Ethernet SLA provides flexibility to specify which performance parameters should be collected and stored for each measurement probe. Performance parameters include frame delay and jitter (inter-frame delay variation). For each performance parameter, either each individual result can be stored, or the results can be aggregated by storing a counter of the number of results that fall within a particular range. A configurable amount of historical data can also be stored as well as the latest results.
• Analyzing and displaying results.
Ethernet SLA performs some basic statistical analysis on the collected results, such as calculating the minimum, maximum, mean and standard deviation. It also records whether any of the probe packets were lost or misordered, or if there is any reason why the results may not be a true reflection of the performance (for example if a big jump in the local time-of-day clock was detected during the time when the measurements were being made).

**Ethernet SLA Concepts**

To successfully configure the Cisco Ethernet SLA feature, you should understand the following concepts:

- Ethernet SLA Statistic, page 90
- Ethernet SLA Measurement Packet, page 90
- Ethernet SLA Sample, page 91
- Ethernet SLA Probe, page 91
- Ethernet SLA Burst, page 91
- Ethernet SLA Schedule, page 91
- Ethernet SLA Bucket, page 91
- Ethernet SLA Aggregation Bin, page 92
- Ethernet SLA Operation Profile, page 92
- Ethernet SLA Operation, page 92
- Ethernet SLA On-Demand Operation, page 92

**Ethernet SLA Statistic**

A *statistic* in Ethernet SLA is a single performance parameter. The following statistics can be measured by Ethernet SLA:

- Round-trip delay
- Round-trip jitter
- One-way delay from source to destination
- One-way jitter from source to destination
- One-way delay from destination to source
- One-way jitter from destination to source

**Note**

Not all statistics can be measured by all types of packet. For example, one-way statistics cannot be measured when using CFM loopback packets.

**Ethernet SLA Measurement Packet**

An Ethernet SLA *measurement packet* is a single protocol message and corresponding reply that is sent on the network for the purpose of making SLA measurements. The following types of measurement packet are supported:
- CFM Delay Measurement (Y.1731 DMM/DMR packets)—CFM delay measurement packets contain timestamps within the packet data that can be used for accurate measurement of frame delay and jitter. These packets can be used to measure round-trip or one-way statistics; however, the size of the DMM/DMR packets cannot be modified.
- CFM loopback (LBM/LBR)—CFM loopback packets are less accurate, but can be used if the peer device does not support DMM/DMR packets. Only round-trip statistics can be measured because these packets do not contain timestamps. However, loopback packets can be padded, so measurements can be made using frames of a specific size.

**Ethernet SLA Sample**

A *sample* is a single result—a number—that relates to a given statistic. For some statistics such as round-trip delay, a sample can be measured using a single measurement packet. For other statistics such as jitter, obtaining a sample requires two measurement packets.

**Ethernet SLA Probe**

A *probe* is a sequence of measurement packets used to gather SLA samples for a specific set of statistics. The measurement packets in a probe are of a specific type (for example, CFM delay measurement or CFM loopback) and have specific attributes, such as the frame size and priority.

<table>
<thead>
<tr>
<th>Note</th>
<th>A single probe can collect data for different statistics at the same time, using the same measurement packets (for example, one-way delay and round-trip jitter).</th>
</tr>
</thead>
</table>

**Ethernet SLA Burst**

Within a probe, measurement packets can either be sent individually, or in bursts. A *burst* contains two or more packets sent within a short interval apart. Each burst can last up to one minute, and bursts can follow each other immediately to provide continuous measurement within the probe.

For statistics that require two measurement packets for each sample (such as jitter), samples are only calculated based on measurement packets in the same burst. For all statistics, it is more efficient to use bursts than to send individual packets.

**Ethernet SLA Schedule**

An Ethernet SLA *schedule* describes how often probes are sent, how long each probe lasts, and at what time the first probe starts.

**Ethernet SLA Bucket**

For a particular statistic, a *bucket* is a collection of results that were gathered during a particular period of time. All of the samples for measurements that were initiated during the period of time represented by a bucket are stored in that bucket. Buckets allow results from different periods of time to be compared (for example, peak traffic to off-peak traffic).

By default, a separate bucket is created for each probe; that is, the bucket represents the period of time starting at the same time as the probe started, and continuing for the duration of the probe. The bucket will therefore contain all the results relating to measurements made by that probe.
Ethernet SLA Aggregation Bin

Rather than storing each sample separately within a bucket, an alternative is to aggregate the samples into bins. An *aggregation bin* is a range of sample values, and contains a counter of the number of samples that were received that fall within that range. The set of bins forms a histogram. When aggregation is enabled, each bucket contains a separate set of bins. See Figure 14 on page 178.

Ethernet SLA Operation Profile

An *operation profile* is a configuration entity that defines the following aspects of an operation:

- What packet types to send and in what quantities (probe and burst configuration)
- What statistics to measure, and how to aggregate them
- When to schedule the probes

An operation profile by itself does not cause any packets to be sent or statistics collected, but is used to create operation instances.

Ethernet SLA Operation

An *operation* is an instance of a given operation profile that is actively collecting performance data. Operation instances are created by associating an operation profile with a given source (an interface and MEP) and with a given destination (a MEP ID or MAC address). Operation instances exist for as long as the configuration is applied, and they run for an indefinite duration on an ongoing basis.

Ethernet SLA On-Demand Operation

An *on-demand operation* is a method of Ethernet SLA operation that can be run on an as-needed basis for a specific and finite period of time. This can be useful in situations such as when you are starting a new service or modifying the parameters for a service to verify the impact of the changes, or if you want to run a more detailed probe when a problem is detected by an ongoing scheduled operation.

On-demand operations do not use profiles and have a finite duration. The statistics that are collected are discarded after a finite time after the operation completes (two weeks), or when you manually clear them. On-demand operations are not persistent so they are lost during certain events such as a card reload or Minimal Disruptive Restart (MDR).

Statistics Measurement and Ethernet SLA Operations Overview

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.

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

- Packet loss count
- Packet corruption event
- Out-of-order event
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 and jitter statistics, the minimum, maximum, mean and standard deviation for the whole bucket are reported, as well as the individual samples or aggregated bins.

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.

By default, there is a separate bucket for each probe. The time period is determined by how long the probe lasts (configured by the `probe`, `send (SLA)`, and `schedule (SLA)` commands). You can modify the size of buckets so that you can have more buckets per probe or fewer buckets per probe (less buckets allows the results from multiple probes to be included in the same bucket). Changing the size of the buckets for a given metric clears all stored data for that metric. All existing buckets are deleted and new buckets are created.

Scheduled SLA operation profiles run indefinitely, according to a configured schedule, and the statistics that are collected are stored in a rolling buffer, where data in the oldest bucket is discarded when a new bucket needs to be recorded.

### Configuration Overview of Scheduled Ethernet SLA Operations

When you configure a scheduled Ethernet SLA operation, you perform the following basic steps:

1. Configure global profiles to define how packets are sent in each probe, how the probes are scheduled, and how the results are stored.
2. Configure operations from a specific local MEP to a specific peer MEP using these profiles.

**Note**

Certain Ethernet SLA configurations use large amounts of memory which can affect the performance of other features on the system. For more information, see the “Configuring Ethernet SLA” section on page 131.

### Ethernet LMI

The Cisco ASR 9000 Series Router supports the Ethernet Local Management Interface (E-LMI) protocol as defined by the *Metro Ethernet Forum, Technical Specification MEF 16, Ethernet Local Management Interface (E-LMI), January 2006* standard.
E-LMI runs on the link between the customer-edge (CE) device and the provider-edge (PE) device, or User Network Interface (UNI), and provides a way for the CE device to auto-configure or monitor the services offered by the PE device (see Figure 13).

**Figure 13** E-LMI Communication on CE-to-PE Link

E-LMI is an asymmetric protocol whose basic operation involves the User-facing PE (uPE) device providing connectivity status and configuration parameters to the CE using STATUS messages in response to STATUS ENQUIRY messages sent by the CE to the uPE.

**E-LMI Messaging**

The E-LMI protocol as defined by the MEF 16 standard, defines the use of only two message types—STATUS ENQUIRY and STATUS. These E-LMI messages consist of required and optional fields called information elements, and all information elements are associated with assigned identifiers. All messages contain the Protocol Version, Message Type, and Report Type information elements, followed by optional information elements and sub-information elements.

E-LMI messages are encapsulated in 46- to 1500-byte Ethernet frames, which are based on the IEEE 802.3 untagged MAC-frame format. E-LMI frames consist of the following fields:

- Destination address (6 bytes)—Uses a standard MAC address of 01:80:C2:00:00:07.
- Source address (6 bytes)—MAC address of the sending device or port.
- E-LMI Ethertype (2 bytes)—Uses 88-EE.
- E-LMI PDU (46–1500 bytes)—Data plus 0x00 padding as needed to fulfill minimum 46-byte length.
- CRC (4 bytes)—Cyclic Redundancy Check for error detection.
For more details about E-LMI messages and their supported information elements, refer to the Metro Ethernet Forum, Technical Specification MEF 16, Ethernet Local Management Interface (E-LMI), January 2006.

**Cisco-Proprietary Remote UNI Details Information Element**

The E-LMI MEF 16 specification does not define a way to send proprietary information.

To provide additional information within the E-LMI protocol, the Cisco IOS XR software implements a Cisco-proprietary information element called Remote UNI Details to send information to the CE about remote UNI names and states. This information element implements what is currently an unused identifier from the E-LMI MEF 16 specification.

To ensure compatibility for future implementations of E-LMI should this identifier ever be implemented in the standard protocol, or for another reason, you can disable transmission of the Remote UNI information element using the `extension remote-uni disable` command.

**E-LMI Operation**

The basic operation of E-LMI consists of a CE device sending periodic STATUS ENQUIRY messages to the PE device, followed by mandatory STATUS message responses by the PE device that contain the requested information. Sequence numbers are used to correlate STATUS ENQUIRY and STATUS messages between the CE and PE.

The CE sends the following two forms of STATUS ENQUIRY messages called Report Types:

- **E-LMI Check**—Verifies a Data Instance (DI) number with the PE to confirm that the CE has the latest E-LMI information.
- **Full Status**—Requests information from the PE about the UNI and all EVCs.

The CE device uses a polling timer to track sending of STATUS ENQUIRY messages, while the PE device can optionally use a Polling Verification Timer (PVT), which specifies the allowable time between transmission of the PE’s STATUS message and receipt of a STATUS ENQUIRY from the CE device before recording an error.

In addition to the periodic STATUS ENQUIRY/STATUS message sequence for the exchange of E-LMI information, the PE device also can send asynchronous STATUS messages to the CE device to communicate changes in EVC status as soon as they occur and without any prompt by the CE device to send that information.

Both the CE and PE devices use a status counter (N393) to determine the local operational status of E-LMI by tracking consecutive errors received before declaring a change in E-LMI protocol status.

**Supported E-LMI PE Functions on the Cisco ASR 9000 Series Router**

The Cisco ASR 9000 Series Router serves as the PE device for E-LMI on a MEN, and supports the following PE functions:

- Supports the E-LMI protocol on Ethernet physical interfaces that are configured with Layer 2 subinterfaces as Ethernet Flow Points (EFPs), which serve as the EVCs about which the physical interface reports status to the CE. The Cisco IOS XR software does not support a specific manageability context for an Ethernet Virtual Connection (EVC).
For E-LMI on the Cisco ASR 9000 Series Router, the term EVC in this documentation refers to a Layer 2 subinterface/EFP.

- Provides the ability to configure the following E-LMI options defined in the MEF 16 specification:
  - T392 Polling Verification Timer (PVT)
  - N393 Status Counter
- Sends notification of the addition and deletion of an EVC.
- Sends notification of the availability (active) or unavailability (inactive, partially active) status of a configured EVC.
- Sends notification of the local UNI name.
- Sends notification of remote UNI names and states using the Cisco-proprietary Remote UNI Details information element, and the ability to disable the Cisco-proprietary Remote UNI information element.
- Sends information about UNI and EVC attributes to the CE (to allow the CE to auto-configure these attributes), including:
  - CE-VLAN to EVC Map
  - CE-VLAN Map Type (Bundling, All-to-one Bundling, Service Multiplexing)
  - Service Type (point-to-point or multipoint)
- Uses CFM Up MEPs to retrieve the EVC state, EVC Service Type, and remote UNI details.
- Provides the ability to retrieve the per-interface operational state of the protocol (including all the information currently being communicated by the protocol to the CE) using the command-line interface (CLI) or Extensible Markup Language (XML) interface.
- Supports up to 80 E-LMI sessions per linecard (one per physical interface).
- Supports up to 32000 EVCs total per linecard for all physical interfaces enabled for E-LMI.

Unsupported E-LMI Functions

These areas of E-LMI are not supported on the Cisco ASR 9000 Series Router:

- CE functions
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.

UDLD Operation

UDLD works by exchanging protocol packets between the neighboring devices. In order for UDLD to work, both devices on the link must support UDLD and have it enabled on respective ports.

UDLD sends an initial PROBE message on the ports where it is configured. Once UDLD receives a PROBE message, it sends periodic ECHO (hello) messages. Both messages identify the sender and its port, and also contain some information about the operating parameters of the protocol on that port. They also contain the device and port identifiers for any neighbor devices that the local device has heard from, on the port. Similarly, each device gets to know where it is connected and where its neighbors are connected.

This information can then be used to detect faults and miswiring conditions. The protocol operates an aging mechanism by means of which information from neighbors that is not periodically refreshed is eventually timed out. This mechanism can also be used for fault detection.

A FLUSH message is used to indicate that UDLD is disabled on a port, which causes the peers to remove the local device from their neighbor cache, to prevent it from being aged out.

If a problem is detected, UDLD disables the affected interface and also notifies the user. This is to avoid further network problems beyond traffic loss, such as loops which are not detected or prevented by STP.

Types of Fault Detection

UDLD can detect these types of faults:

- **Transmit faults** — These are cases where there has been a failure in transmitting packets from the local port to the peer device, but packets continue to be 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 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, page 98.
UDLD Modes of Operation

UDLD can operate in these modes:

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

UDLD Aging Mechanism

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

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

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

State Machines

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

Main FSM

The Main FSM can be in one of these states:

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

The Detection FSM can be in one of these states:

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

Unidirectional Link Routing

Unidirectional Link Routing (UDLR) feature allows a port to unidirectionally transmit or receive traffic. Therefore, instead of using two strands of fiber for a full-duplex Gigabit Ethernet or 10Gigabit Ethernet port, UDLR uses only one strand of fiber that either transmits or receives the one-way traffic depending on the configuration. This improves the effectiveness and also enables you to double the bandwidth with existing fiber infrastructure.

Cisco IOS XR Software supports Unidirectional Link Routing (UDLR) feature on these line cards:

- A9K-24T-TR 24-port 10 Gigabit Ethernet line cards
- A9K-24T-SE 24-port 10 Gigabit Ethernet line cards
- A9K-36T-TR 36-port 10 Gigabit Ethernet line cards
- A9K-36T-SE 36-port 10 Gigabit Ethernet line cards

UDLR is used for applications such as video streaming, where most of the traffic is sent as unacknowledged unidirectional video broadcast streams.

How to Configure Ethernet OAM

This section provides these configuration procedures:

- Configuring Ethernet Link OAM, page 100
- Configuring Ethernet CFM, page 108
- Configuring Ethernet SLA, page 131
- Configuring Ethernet LMI, page 141
- Configuring UDLD, page 159
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, page 100
- Attaching an Ethernet OAM Profile to an Interface, page 105
- Configuring Ethernet OAM at an Interface and Overriding the Profile Configuration, page 106
- Verifying the Ethernet OAM Configuration, page 108

Configuring an Ethernet OAM Profile

Perform the following 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
6. frame window window
7. frame threshold low threshold
8. frame-period window window
9. frame-period threshold low threshold
10. frame-seconds window window
11. frame-seconds threshold low threshold
12. exit
13. mib-retrieval
14. connection timeout seconds
15. hello-interval {100ms | 1s}
16. mode {active | passive}
17. require-remote mode {active | passive}
18. require-remote link-monitoring
19. require-remote mib-retrieval
20. action capabilities-conflict {disable | efd | error-disable-interface}
21. action critical-event {disable | error-disable-interface}
22. `action discovery-timeout {disable | efd | error-disable-interface}`
23. `action dying-gasp {disable | error-disable-interface}`
24. `action high-threshold {error-disable-interface | log}`
25. `action remote-loopback disable`
26. `action session-down {disable | efd | error-disable-interface}`
27. `action session-up disable`
28. `action uni-directional link-fault {disable | efd | error-disable-interface}`
29. `action wiring-conflict {disable | efd | log}`
30. `uni-directional link-fault detection`
31. `commit`
32. `end`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>Configure</strong></td>
</tr>
<tr>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router# configure terminal</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>Ethernet OAM Profile</strong></td>
</tr>
<tr>
<td><code>ethernet oam profile profile-name</code></td>
<td>Creates a new Ethernet Operations, Administration and Maintenance (OAM) profile and enters Ethernet OAM configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router(config)# ethernet oam profile Profile_1</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>Link Monitor</strong></td>
</tr>
<tr>
<td><code>link-monitor</code></td>
<td>Enters the Ethernet OAM link monitor configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router(config-eoam)# link-monitor</code></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>Symbol Period</strong></td>
</tr>
<tr>
<td><code>symbol-period window window</code></td>
<td>(Optional) Configures the window size (in milliseconds) for an Ethernet OAM symbol-period error event.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router(config-eoam-lm)# symbol-period window 60000</code></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>Symbol Period Threshold</strong></td>
</tr>
<tr>
<td><code>symbol-period threshold low threshold high threshold</code></td>
<td>(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.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router(config-eoam-lm)# symbol-period threshold low 10000000 high 60000000</code></td>
</tr>
</tbody>
</table>
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td><code>frame window window</code></td>
<td>(Optional) Configures the frame window size (in milliseconds) of an OAM frame error event. The range is 1000 to 60000. The default value is 1000.</td>
</tr>
<tr>
<td>7</td>
<td><code>frame threshold low threshold high threshold</code></td>
<td>(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. The range is 0 to 60000000. The default low threshold is 1.</td>
</tr>
<tr>
<td>8</td>
<td><code>frame-period window window</code></td>
<td>(Optional) Configures the window size (in milliseconds) for an Ethernet OAM frame-period error event. The range is 100 to 60000. The default value is 1000.</td>
</tr>
<tr>
<td>9</td>
<td><code>frame-period threshold low threshold high threshold</code></td>
<td>(Optional) Configures the thresholds (in frames) that trigger an Ethernet OAM frame-period error event. The high threshold is optional and is configurable only in conjunction with the low threshold. The range is 0 to 1000000. The default low threshold is 60000.</td>
</tr>
<tr>
<td>10</td>
<td><code>frame-seconds window window</code></td>
<td>(Optional) Configures the window size (in milliseconds) for the OAM frame-seconds error event. The range is 10000 to 900000. The default value is 6000.</td>
</tr>
<tr>
<td>11</td>
<td><code>frame-seconds threshold low threshold high threshold</code></td>
<td>(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.</td>
</tr>
<tr>
<td>12</td>
<td><code>exit</code></td>
<td>Exits back to Ethernet OAM mode.</td>
</tr>
<tr>
<td>13</td>
<td><code>mib-retrieval</code></td>
<td>Enables MIB retrieval in an Ethernet OAM profile or on an Ethernet OAM interface.</td>
</tr>
</tbody>
</table>

### Example:

- **Step 6:**
  ```
  RP/0/RSP0/CPU0:router(config-eoam-lm)# frame window 60
  ```

- **Step 7:**
  ```
  RP/0/RSP0/CPU0:router(config-eoam-lm)# frame threshold low 10000000 high 60000000
  ```

- **Step 8:**
  ```
  RP/0/RSP0/CPU0:router(config-eoam-lm)# frame-period window 60000
  ```

- **Step 9:**
  ```
  RP/0/RSP0/CPU0:router(config-eoam-lm)# frame-period threshold low 100 high 1000000
  ```

- **Step 10:**
  ```
  RP/0/RSP0/CPU0:router(config-eoam-lm)# frame-seconds window 900000
  ```

- **Step 11:**
  ```
  RP/0/RSP0/CPU0:router(config-eoam-lm)# frame-seconds threshold low 3 threshold 900
  ```

- **Step 12:**
  ```
  RP/0/RSP0/CPU0:router(config-eoam-lm)# exit
  ```

- **Step 13:**
  ```
  RP/0/RSP0/CPU0:router(config-eoam)# mib-retrieval
  ```
### Command or Action

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>connection timeout seconds</td>
<td>Configures the timeout value (in seconds) for an Ethernet OAM session. The range is 2 to 30. The default value is 5.</td>
</tr>
<tr>
<td>15</td>
<td>hello-interval {100ms</td>
<td>1s}</td>
</tr>
<tr>
<td>16</td>
<td>mode {active</td>
<td>passive}</td>
</tr>
<tr>
<td>17</td>
<td>require-remote mode {active</td>
<td>passive}</td>
</tr>
<tr>
<td>18</td>
<td>require-remote link-monitoring</td>
<td>Requires that link-monitoring is configured on the remote end before the OAM session becomes active.</td>
</tr>
<tr>
<td>19</td>
<td>require-remote mib-retrieval</td>
<td>Requires that MIB-retrieval is configured on the remote end before the OAM session becomes active.</td>
</tr>
<tr>
<td>20</td>
<td>action capabilities-conflict (disable</td>
<td>efd</td>
</tr>
<tr>
<td>21</td>
<td>action critical-event (disable</td>
<td>error-disable-interface)</td>
</tr>
</tbody>
</table>
**Configure Ethernet OAM on the Cisco ASR 9000 Series Router**

**How to Configure Ethernet OAM**

**Step 22**

**Command or Action**

```
action discovery-timeout (disable | efd | error-disable-interface)
```

**Example**:

```
RP/0/RSP0/CPU0:router(config-eoam)# action discovery-timeout efd
```

**Purpose**

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 23**

**Command or Action**

```
action dying-gasp (disable | error-disable-interface)
```

**Example**:

```
RP/0/RSP0/CPU0:router(config-eoam)# action dying-gasp error-disable-interface
```

**Purpose**

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.

**Step 24**

**Command or Action**

```
action high-threshold (error-disable-interface | log)
```

**Example**:

```
RP/0/RSP0/CPU0:router(config-eoam)# action high-threshold error-disable-interface
```

**Purpose**

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 25**

**Command or Action**

```
action remote-loopback disable
```

**Example**:

```
RP/0/RSP0/CPU0:router(config-eoam)# action remote-loopback disable
```

**Purpose**

Specifies that no action is taken on an interface when a remote-loopback 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 26**

**Command or Action**

```
action session-down (disable | efd | error-disable-interface)
```

**Example**:

```
RP/0/RSP0/CPU0:router(config-eoam)# action session-down efd
```

**Purpose**

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 27**

**Command or Action**

```
action session-up disable
```

**Example**:

```
RP/0/RSP0/CPU0:router(config-eoam)# action session-up disable
```

**Purpose**

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.
## Configuring Ethernet OAM on the Cisco ASR 9000 Series Router

**How to Configure Ethernet OAM**

### Attaching an Ethernet OAM Profile to an Interface

Perform the following steps to attach an Ethernet OAM profile to an interface:

**SUMMARY STEPS**

1. configure
2. interface GigabitEthernet interface-path-id
3. ethernet oam
4. profile profile-name
5. commit
6. end

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 28</strong> action uni-directional link-fault (disable</td>
<td>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.</td>
</tr>
<tr>
<td>efd</td>
<td><strong>Note</strong> If you change the default, the <strong>log</strong> 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.</td>
</tr>
<tr>
<td>error-disable-interface)</td>
<td></td>
</tr>
<tr>
<td><strong>Step 29</strong> action wiring-conflict (disable</td>
<td>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.</td>
</tr>
<tr>
<td>efd</td>
<td><strong>Note</strong> If you change the default, the <strong>error-disable-interface</strong> 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.</td>
</tr>
<tr>
<td>log)</td>
<td></td>
</tr>
<tr>
<td><strong>Step 30</strong> uni-directional link-fault detection</td>
<td>Enables detection of a local, unidirectional link fault and sends notification of that fault to an Ethernet OAM peer.</td>
</tr>
<tr>
<td><strong>Step 31</strong> commit</td>
<td>Saves the configuration changes to the running configuration file and remains within the configuration session.</td>
</tr>
<tr>
<td><strong>Step 32</strong> end</td>
<td>Ends the configuration session and exits to the EXEC mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config-eoam)# action session-down efd
```

```
RP/0/RSP0/CPU0:router(config-eoam)#
uni-directional link-fault detection
```

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

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface GigabitEthernet interface-path-id</td>
<td>Enters interface configuration mode and specifies the Ethernet interface name and notation rack/slot/module/port.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/1/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ethernet oam</td>
<td>Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if)# ethernet oam</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> profile profile-name</td>
<td>Attaches the specified Ethernet OAM profile (profile-name), and all of its configuration, to the interface.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if-eoam)# profile Profile_1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> commit</td>
<td>Saves the configuration changes to the running configuration file and remains within the configuration session.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if)# commit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> end</td>
<td>Ends the configuration session and exits to the EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if)# end</td>
<td></td>
</tr>
</tbody>
</table>

#### Configuring Ethernet OAM at an Interface and Overriding the Profile Configuration

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

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

To see all of the default Ethernet OAM configuration settings, see the “Verifying the Ethernet OAM Configuration” section on page 108.
To configure Ethernet OAM settings at an interface and override the profile configuration, perform the following steps:

SUMMARY STEPS

1. configure
2. interface GigabitEthernet interface-path-id
3. ethernet oam
4. interface-Ethernet-OAM-command
5. commit
6. end

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface GigabitEthernet interface-path-id</td>
<td>Enters interface configuration mode and specifies the Ethernet interface name and notation rack/slot/module/port.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet 0/1/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ethernet oam</td>
<td>Enables Ethernet OAM and enters interface Ethernet OAM configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if)# ethernet oam</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> interface-Ethernet-OAM-command</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if-eoam)# action capabilities-conflict error-disable-interface</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> commit</td>
<td>Saves the configuration changes to the running configuration file and remains within the configuration session.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if)# commit</td>
<td></td>
</tr>
<tr>
<td><strong>Step 6</strong> end</td>
<td>Ends the configuration session and exits to the EXEC mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if)# end</td>
<td></td>
</tr>
</tbody>
</table>
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/RSP0/CPU0:router# show ethernet oam configuration
Thu Aug 5 22:07:06.870 DST
GigabitEthernet0/4/0/0:
  Hello interval:                                              1s
  Link monitoring enabled:                                      Y
  Remote loopback enabled:                                     N
  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
  Remote loopback action:                                      Log
  Require remote mode:                                        Ignore
  Require remote MIB retrieval:                                N
  Require remote loopback support:                            N
  Require remote link monitoring:                             N
```

Configuring Ethernet CFM

To configure Ethernet CFM, perform the following tasks:

- Configuring a CFM Maintenance Domain, page 109 (required)
- Configuring Services for a CFM Maintenance Domain, page 110 (required)
- Enabling and Configuring Continuity Check for a CFM Service, page 112 (optional)
- Configuring Automatic MIP Creation for a CFM Service, page 114 (optional)
- Configuring Cross-Check on a MEP for a CFM Service, page 116 (optional)
- Configuring Other Options for a CFM Service, page 118 (optional)
- Configuring CFM ME Ps, page 120 (required)
- Configuring Y.1731 AIS, page 122 (optional)
Configuring a CFM Maintenance Domain

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

**SUMMARY STEPS**

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

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2 ethernet cfm</td>
<td>Enters Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td>Step 3 domain domain-name level level-value [id [null]</td>
<td>Creates and names a container for all domain configurations and enters</td>
</tr>
<tr>
<td>[dns DNS-name] [mac H.H.H] [string string] ]</td>
<td>CFM domain configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-cfm)# domain</td>
<td>Creates and names a container for all domain configurations and enters</td>
</tr>
<tr>
<td>Domain_One level 1 id string D1</td>
<td>CFM domain configuration mode.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.
### How to Configure Ethernet OAM

#### 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 {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. `end`
   or
   `commit`

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td>(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.</td>
</tr>
<tr>
<td><code>traceroute cache hold-time minutes size entries</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-cfm)# traceroute cache hold-time 1 size 3000</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><code>end</code> or <code>commit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-cfm-dmn)# commit</code></td>
<td></td>
</tr>
</tbody>
</table>

---

#### 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 {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. `end`
   or
   `commit`

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td>(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.</td>
</tr>
<tr>
<td><code>traceroute cache hold-time minutes size entries</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-cfm)# traceroute cache hold-time 1 size 3000</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><code>end</code> or <code>commit</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>RP/0/RSP0/CPU0:router(config-cfm-dmn)# commit</code></td>
<td></td>
</tr>
</tbody>
</table>
**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>configure</strong></td>
<td></td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

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

<table>
<thead>
<tr>
<th>Step 2</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ethernet cfm</strong></td>
<td></td>
<td>Enters Ethernet CFM configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config)# ethernet cfm
```

<table>
<thead>
<tr>
<th>Step 3</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string] ]</strong></td>
<td>Creates and names a container for all domain configurations at a specified maintenance level, and enters CFM domain configuration mode.</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1
```

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.
### How to Configure Ethernet OAM

The Cisco ASR 9000 Series Router supports Continuity Check as defined in the IEEE 802.1ag specification, and supports CCMs intervals of 100 ms and longer. The overall packet rates for CCM messages are up to 16000 CCMs-per-second sent, and up to 16000 CCMs-per-second received, per card.

**Note**

If Ethernet SLA is configured, the overall combined packet rate for CCMs and SLA frames is 16000 frames-per-second in each direction, per card.

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]]`

---

### Enabling and Configuring Continuity Check for a CFM Service

The Cisco ASR 9000 Series Router supports Continuity Check as defined in the IEEE 802.1ag specification, and supports CCMs intervals of 100 ms and longer. The overall packet rates for CCM messages are up to 16000 CCMs-per-second sent, and up to 16000 CCMs-per-second received, per card.

**Note**

If Ethernet SLA is configured, the overall combined packet rate for CCMs and SLA frames is 16000 frames-per-second in each direction, per card.

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]]`

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **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]}
```

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.

**Example:**

RP/0/RSP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1

**Step 5**

```
end
```

or

```
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)?
  
  - 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.

**Example:**

RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# commit
## Configuring Ethernet OAM on the Cisco ASR 9000 Series Router

**How to Configure Ethernet OAM**

**HC-113**

**Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide**

**OL-26061-03**

1. **Step 1**
   - **Command or Action**: configure
   - **Example**: RP/0/RSP0/CPU0:router# configure
   - **Purpose**: Enters global configuration mode.

2. **Step 2**
   - **Command or Action**: ethernet cfm
   - **Example**: RP/0/RSP0/CPU0:router(config)# ethernet cfm
   - **Purpose**: Enters Ethernet Connectivity Fault Management (CFM) configuration mode.

3. **Step 3**
   - **Command or Action**: domain domain-name level level-value [id] [null] [dns DNS-name] [mac H.H.H] [string string]
   - **Example**: RP/0/RSP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1
   - **Purpose**: Creates and names a container for all domain configurations and enters the CFM domain configuration mode.
   - **Details**: 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.

4. **Step 4**
   - **Command or Action**: 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/RSP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1
   - **Purpose**: 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.
   - **Details**: The id sets the short MA name.

5. **Step 5**
   - **Command or Action**: continuity-check interval time [loss-threshold threshold]
   - **Example**: RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# continuity-check interval 100m loss-threshold 10
   - **Purpose**: (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.

---

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>2</td>
<td>ethernet cfm</td>
<td>Enters Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td>3</td>
<td>domain domain-name level level-value [id] [null] [dns DNS-name] [mac H.H.H] [string string]</td>
<td>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.</td>
</tr>
<tr>
<td>4</td>
<td>service service-name [bridge group bridge-domain-group bridge-domain] bridge-domain-name</td>
<td>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.</td>
</tr>
<tr>
<td>5</td>
<td>continuity-check interval time [loss-threshold threshold]</td>
<td>(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.</td>
</tr>
</tbody>
</table>
### How to Configure Ethernet OAM

**Configuring Automatic MIP Creation for a CFM Service**

For more information about the algorithm for creating MIPs, see the “MIP Creation” section on page 74.

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 [bridge group bridge-domain-group bridge-domain bridge-domain-name | down-meps | xconnect group xconnect-group-name p2p xconnect-name][id [icc-based ice-string umc-string] | [string text] | [number number] | [vlan-id id-number] | [vpn-id oui-vpnid]]`
5. `mip auto-create {all | lower-mep-only}`

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 6</td>
<td><code>continuity-check archive hold-time minutes</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# continuity-check archive hold-time 100</td>
</tr>
<tr>
<td>Step 7</td>
<td><code>continuity-check loss auto-traceroute</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# continuity-check loss auto-traceroute</td>
</tr>
<tr>
<td>Step 8</td>
<td><code>end</code> or <code>commit</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# commit</td>
</tr>
</tbody>
</table>

- 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.
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>ethernet cfm</code></td>
<td>Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>domain domain-name level level-value [id [null] [dns DNS-name] [mac H.H.H] [string string] ]</code></td>
<td>Creates and names a container for all domain configurations and enters the CFM domain configuration mode. The level must be specified. The <code>id</code> 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.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> `service service-name {bridge group bridge-domain-group bridge-domain bridge-domain-name</td>
<td>down-meps</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1</td>
<td></td>
</tr>
</tbody>
</table>
## 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`

### Command or Action | Purpose
---|---
**Step 5** `mip auto-create (all | lower-mep-only)` | (Optional) Enables the automatic creation of MIPs in a bridge domain or xconnect.

**Example:**
```
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# mip auto-create all
```

**Step 6** `end`
   or
   `commit`

**Example:**
```
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# commit
```

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 Ethernet OAM on the Cisco ASR 9000 Series Router

## How to Configure Ethernet OAM

### Detailed Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>ethernet cfm</strong></td>
<td>Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router# ethernet cfm</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>domain</strong></td>
<td>Creates and names a container for all domain configurations and enters the CFM domain configuration mode.</td>
</tr>
<tr>
<td><strong>domain-name</strong></td>
<td><em>level</em></td>
<td>The level must be specified.</td>
</tr>
<tr>
<td><strong>level-value</strong></td>
<td><em>id</em></td>
<td>The <em>id</em> 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.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>service</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>service-name</strong></td>
<td><em>bridge group</em></td>
<td>The <em>id</em> sets the short MA name.</td>
</tr>
<tr>
<td><strong>bridge-domain-group</strong></td>
<td><em>bridge-domain</em></td>
<td></td>
</tr>
<tr>
<td><strong>bridge-domain-name</strong></td>
<td><em>down-meps</em></td>
<td></td>
</tr>
<tr>
<td><strong>xconnect group</strong></td>
<td><em>xconnect-group-name</em></td>
<td></td>
</tr>
<tr>
<td><strong>xconnect-name</strong></td>
<td><em>icc-based</em></td>
<td></td>
</tr>
<tr>
<td><strong>icc-string</strong></td>
<td><em>umc-string</em></td>
<td></td>
</tr>
<tr>
<td><strong>text</strong></td>
<td><em>number</em></td>
<td></td>
</tr>
<tr>
<td><strong>number</strong></td>
<td><em>vlan-id</em></td>
<td></td>
</tr>
<tr>
<td><strong>id-number</strong></td>
<td><em>vpn-id</em></td>
<td></td>
</tr>
<tr>
<td><strong>oui-vpnid</strong></td>
<td><em>string</em></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>mep crosscheck</strong></td>
<td>Enters CFM MEP crosscheck configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# mep crosscheck mep-id 10</code></td>
<td></td>
</tr>
</tbody>
</table>
### 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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 6 mep-id mep-id-number [mac-address mac-address]</td>
<td>Enables cross-check on a MEP.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-cfm-xcheck)# mep-id 10</td>
<td>Note Repeat this command for every MEP that you want included in the expected set of MEPs for cross-check.</td>
</tr>
<tr>
<td>Step 7 end or commit</td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>

- 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.

**Example:**

```
RP/0/RSP0/CPU0:router(config-cfm-xcheck)# mep-id 10
Enables cross-check on a MEP.

Example:
RP/0/RSP0/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.

```
Example:
RP/0/RSP0/CPU0:router(config-cfm-xcheck)# commit
```

---

**Command or Action**

**Purpose**

| Step 6 mep-id mep-id-number [mac-address mac-address] | Enables cross-check on a MEP. |
| Example: RP/0/RSP0/CPU0:router(config-cfm-xcheck)# mep-id 10 | 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 | 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.

```
Example:
RP/0/RSP0/CPU0:router(config-cfm-xcheck)# mep-id 10
Enables cross-check on a MEP.

Example:
RP/0/RSP0/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.

```
Example:
RP/0/RSP0/CPU0:router(config-cfm-xcheck)# commit
```

---

**Command or Action**

**Purpose**

| Step 6 mep-id mep-id-number [mac-address mac-address] | Enables cross-check on a MEP. |
| Example: RP/0/RSP0/CPU0:router(config-cfm-xcheck)# mep-id 10 | 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 | 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.

```
Example:
RP/0/RSP0/CPU0:router(config-cfm-xcheck)# mep-id 10
Enables cross-check on a MEP.

Example:
RP/0/RSP0/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.

```
Example:
RP/0/RSP0/CPU0:router(config-cfm-xcheck)# commit
```
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> ethernet cfm</td>
<td>Enters the Ethernet Connectivity Fault Management (CFM) configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> domain domain-name level level-value [id null] [dns DNS-name] [mac H.H.H] [string string]</td>
<td>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.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-cfm)# domain Domain_One level 1 id string D1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> service service-name {bridge group bridge-domain-group bridge-domain</td>
<td>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.</td>
</tr>
<tr>
<td>group xconnect-group-name p2p xconnect-name} [id [icc-based icc-string umc-string] [string text] [number number] [vlan-id id-number]</td>
<td></td>
</tr>
<tr>
<td>[vpn-id oui-vpnid]]</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-cfm-dmn)# service Bridge_Service bridge group BD1 bridge-domain B1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> maximum-meps number</td>
<td>(Optional) Configures the maximum number (2 to 8190) of MEPs across the network, which limits the number of peer MEPs recorded in the database.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# maximum-meps 1000</td>
<td></td>
</tr>
</tbody>
</table>
When you configure CFM MEPs, consider the following guidelines:

- Up to 32000 local MEPs are supported per card.
- CFM maintenance points can be created on the following interface types:
  - All physical Ethernet interfaces (except for the RSP Management interfaces).
  - Ethernet bundle interfaces.
  - All physical and bundle Ethernet subinterfaces, providing the encapsulation is configured according to the following guidelines:
    - Frames are only matched based on VLAN IDs and CoS bits.
    - Frames are not matched using VLAN “any.”
  - Ethernet bundle member interfaces—Only down MEPs at level 0 can be created.
- CFM maintenance points can be created on both Layer 2 and Layer 3 interfaces. On L3 interfaces, only down MEPs can be created.
Restrictions

When you configure MEPs, consider the following restrictions:

- Maintenance points at level 0 are not supported on bundle interfaces.
- If a subinterface is configured that matches untagged Ethernet frames (for example, by configuring the `encapsulation default` command), then you cannot create a down MEP on the underlying physical or bundle interface.
- Up MEPs are not supported on Layer 3 interfaces.

SUMMARY STEPS

1. `configure`
2. `interface {GigabitEthernet | TenGigE | Bundle-Ether} interface-path-id.subinterface`
3. `ethernet cfm`
4. `mep domain domain-name service service-name mep-id id-number`
5. `cos cos`
6. `end` or `commit`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> `interface {GigabitEthernet</td>
<td>TenGigE</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>ethernet cfm</code></td>
<td>Enters interface Ethernet CFM configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> <code>RP/0/RSP0/CPU0:router(config-if)# ethernet cfm</code></td>
<td></td>
</tr>
</tbody>
</table>
Configuring Y.1731 AIS

This section has the following step procedures:

- Configuring AIS in a CFM Domain Service
- Configuring AIS on a CFM Interface

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong> mep domain domain-name service service-name mep-id id-number</td>
<td>Creates a maintenance end point (MEP) on an interface and enters interface CFM MEP configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-if-cfm)# mep domain Dm1 service Sv1 mep-id 1</td>
</tr>
<tr>
<td><strong>Step 5</strong> cos cos</td>
<td>(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.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-if-cfm-mep)# cos 7</td>
</tr>
<tr>
<td><strong>Step 6</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-if-cfm-mep)# commit</td>
</tr>
</tbody>
</table>

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.
4. `service name bridge group name bridge-domain name`
5. `ais transmission [interval \{1s | 1m\}] [cos cos]`
6. `log ais`
7. `end`
   or
   `commit`

---

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1 <code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

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

| Step 2 `ethernet cfm` | Enters Ethernet CFM global configuration mode. |

**Example:**
```
RP/0/RSP0/CPU0:router(config)# ethernet cfm
```

| Step 3 `domain name level level` | Specifies the domain and domain level. |

**Example:**
```
RP/0/RSP0/CPU0:router(config-cfm)# domain D1 level 1
```

| Step 4 `service name bridge group name bridge-domain name` | Specifies the service, bridge group, and bridge domain. |

**Example:**
```
RP/0/RSP0/CPU0:router(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2
```

| Step 5 `ais transmission [interval \{1s | 1m\}] [cos cos]` | Configures Alarm Indication Signal (AIS) transmission for a Connectivity Fault Management (CFM) domain service. |

**Example:**
```
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7
```
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
6. `commit`

**Command or Action**

<table>
<thead>
<tr>
<th>Step 6</th>
<th>Command or Action</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>log ais</code></td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# log ais
```

**Purpose:**

Configures AIS logging for a Connectivity Fault Management (CFM) domain service to indicate when AIS or LCK packets are received.

**Step 7**

<table>
<thead>
<tr>
<th>Command or Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>end</code> or <code>commit</code></td>
</tr>
</tbody>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config-sla-prof-stat-cfg)# commit
```

**Purpose:**

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.
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong>&lt;br&gt;Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>configure</code>&lt;br&gt;<code>RP/0/RSP0/CPU0:router# configure</code></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>interface gigabitethernet interface-path-id</strong>&lt;br&gt;Enters interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>interface gigabitethernet 0/1/0/2</code>&lt;br&gt;<code>interface-path-id</code></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>ethernet cfm</strong>&lt;br&gt;Enters Ethernet CFM interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>ethernet cfm</code>&lt;br&gt;<code>RP/0/RSP0/CPU0:router(config)# ethernet cfm</code></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>ais transmission up interval 1m cos cos</strong>&lt;br&gt;Configures Alarm Indication Signal (AIS) transmission on a Connectivity Fault Management (CFM) interface.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>ais transmission up interval 1m cos 7</code>&lt;br&gt;<code>RP/0/RSP0/CPU0:router(config-if-cfm)# ais transmission up interval 1m cos 7</code></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>end</strong> or <strong>commit</strong>&lt;br&gt;Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td><code>end</code> or <code>commit</code>&lt;br&gt;<code>RP/0/RSP0/CPU0:router(config-sla-prof-stat-cfg)# commit</code></td>
</tr>
</tbody>
</table>

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 EFD for a CFM Service

To configure EFD for a CFM service, complete the following steps.

#### Restrictions

EFD is not supported on up MEPs. It can only be configured on down MEPs, within a particular service.

#### SUMMARY STEPS

1. `configure`
2. `ethernet cfm`
3. `domain domain-name level level-value`
4. `service service-name down-meps`
5. `efd`
6. `log efd`
7. `end` or `commit`

#### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> <code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router# configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> <code>ethernet cfm</code></td>
<td>Enters CFM configuration mode.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router(config)# ethernet cfm</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> <code>domain domain-name level level-value</code></td>
<td>Specifies or creates the CFM domain and enters CFM domain configuration mode.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router(config-cfm-dmn)# domain D1 level 1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> <code>service service-name down-meps</code></td>
<td>Specifies or creates the CFM service for down MEPS and enters CFM domain service configuration mode.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router(config-cfm-dmn)# service S1 down-meps</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 5</strong> <code>efd</code></td>
<td>Enables EFD on all down MEPs in the down MEPS service.</td>
</tr>
<tr>
<td>Example: <code>RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# efd</code></td>
<td></td>
</tr>
</tbody>
</table>
Configuring Ethernet OAM on the Cisco ASR 9000 Series Router

How to Configure Ethernet OAM

Verifying the EFD Configuration

The following example shows how to display all interfaces that are shut down because of Ethernet Fault Detection (EFD):

```
RP/0/RSP0/CPU0:router# show efd interfaces
```

**Server VLAN MA**

<table>
<thead>
<tr>
<th>Interface</th>
<th>Clients</th>
</tr>
</thead>
<tbody>
<tr>
<td>GigE0/0/0.0</td>
<td>CFM</td>
</tr>
</tbody>
</table>

Configuring Flexible VLAN Tagging for CFM

Use the following procedure to set the number of tags in CFM packets from up MEPs to 1, 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

(Optional) Enables logging of EFD state changes on an interface.

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.
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# <code>configure</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td><code>ethernet cfm</code></td>
<td>Enters Ethernet CFM global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# <code>ethernet cfm</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td><code>domain name level level</code></td>
<td>Specifies the domain and domain level.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-cfm)# <code>domain D1 level 1</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td><code>service name bridge group name bridge-domain name</code></td>
<td>Specifies the service, bridge group, and bridge domain.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-cfm-dmn)# <code>service S2 bridge group BG1 bridge-domain BD2</code></td>
<td></td>
</tr>
</tbody>
</table>
**Verifying the CFM Configuration**

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td><strong>tags number</strong></td>
<td>Specifies the number of tags in CFM packets from up MEPS. Currently, the only valid value is 1.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# tags 1</td>
<td></td>
</tr>
</tbody>
</table>

| **Step 6** | |
| **end** or **commit** | Saves configuration changes. |
| **Example:** | |
| RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# commit | |

- When you issue the **end** command, the system prompts you to commit changes:

  Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
  
  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.

**Verifying the CFM Configuration**

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

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>show ethernet cfm configuration-errors</strong> [domain domain-name] [interface interface-path-id]</td>
<td>Displays information about errors that are preventing configured CFM operations from becoming active, as well as any warnings that have occurred.</td>
</tr>
<tr>
<td><strong>show ethernet cfm local maintenance-points</strong> domain name [service name] [interface type interface-path-id] [mep</td>
<td>mip]</td>
</tr>
</tbody>
</table>

**Troubleshooting Tips**

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

<table>
<thead>
<tr>
<th>Step 1</th>
<th>To verify connectivity to a problematic MEP, use the <strong>ping ethernet cfm</strong> command as shown in the following example:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# ping ethernet cfm domain D1 service S1 mep-id 16 source interface GigabitEthernet 0/0/0/0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ping ethernet cfm</strong></td>
<td>Ping Ethernet OAM and CFM connectivity to test the reachability of a MEP within the CFM network.</td>
</tr>
</tbody>
</table>
Type escape sequence to abort.
Sending 5 CFM Loopbacks, timeout is 2 seconds -
Domain foo (level 2), Service foo
Source: MEP ID 1, interface GigabitEthernet0/0/0/0
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/RSP0/CPU0:router# traceroute ethernet cfm domain D1 service S1 mep-id 16 source interface gigabitethernet 0/0/0/0
Traceroutes in domain D1 (level 4), service S1
Source: MEP-ID 1, interface GigabitEthernet0/0/0/0
================================================================================================
Traceroute at 2009-05-18 12:09:10 to 0001.0203.0402, TTL 64, Trans ID 2:

<table>
<thead>
<tr>
<th>Hop</th>
<th>Hostname/Last</th>
<th>Ingress MAC/name</th>
<th>Egress MAC/Name</th>
<th>Relay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ios</td>
<td>0001.0203.0400 [Down]</td>
<td></td>
<td>FDB</td>
</tr>
<tr>
<td>2</td>
<td>abc</td>
<td>G10/0/0/0</td>
<td>0001.0203.0401 [Ok]</td>
<td>FDB</td>
</tr>
<tr>
<td>3</td>
<td>bcd</td>
<td>0001.0203.0402 [Ok]</td>
<td></td>
<td>Hit</td>
</tr>
<tr>
<td>abc</td>
<td></td>
<td>GigE0/0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.
Configuring Ethernet SLA

This section describes how to configure Ethernet SLA.

Ethernet SLA 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.
- **The Cisco ASR 9000 Series Router supports SLA packet intervals of 100 ms and longer. If Ethernet SLA is configured, the overall combined packet rate for CCMs and SLA frames is 16000 frames-per-second in each direction, per card.**

The following procedure provides the steps to configure Ethernet Service Level Agreement (SLA) monitoring at Layer 2.

To configure SLA, perform the following tasks:

- Configuring an SLA Operation Profile, page 131
- Configuring SLA Probe Parameters in a Profile, page 132
- Configuring SLA Statistics Measurement in a Profile, page 134
- Configuring a Schedule for an SLA Operation Probe in a Profile, page 136
- Configuring an SLA Operation, page 138
- Configuring an On-Demand SLA Operation, page 139
- Verifying SLA Configuration, page 141

**Configuring an SLA Operation Profile**

To configure a profile, perform the following steps:

**SUMMARY STEPS**

1. `configure`
2. `ethernet sla`
### Configuring Ethernet OAM on the Cisco ASR 9000 Series Router

#### How to Configure Ethernet OAM

3. `profile profile-name type {cfm-delay-measurement | cfm-loopback}
4. `end` or `commit`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure</code> Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><code>ethernet sla</code> Enters the SLA configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# ethernet sla</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>`profile profile-name type (cfm-delay-measurement</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-sla)# profile Prof1 type cfm-loopback</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><code>end</code> or <code>commit</code> Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-sla)# commit</td>
</tr>
</tbody>
</table>

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 SLA Probe Parameters in a Profile

To configure SLA probe parameters in a profile, perform the following steps beginning in SLA profile configuration mode:
### SUMMARY STEPS

1. **probe**
2. **send burst** *(every number {seconds | minutes | hours} | once)* **packet count** **packets** **interval**
   *number* *(seconds | milliseconds)*
   or
   **send packet** *(every number {milliseconds | seconds | minutes | hours} | once)*
3. **packet size** **bytes** *(test pattern (hex 0xHHHHHHHH | pseudo-random))*
4. **priority** **priority**
5. **end**
   or
   **commit**

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><strong>probe</strong></td>
<td>Enters the SLA profile probe configuration mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-sla-prof)# probe</td>
<td></td>
</tr>
</tbody>
</table>
| Step 2 | **send burst** *(every number {seconds | minutes | hours} | once)* **packet count** **packets** **interval**
   *number* *(seconds | milliseconds)*
   or
   **send packet** *(every number {milliseconds | seconds | minutes | hours} | once)*
|        | **Example:** RP/0/RSP0/CPU0:router(config-sla-prof-pb)# send burst every 60 seconds packet count 100 interval 100 milliseconds
   or
   RP/0/RSP0/CPU0:router(config-sla-prof-pb)# send burst once packet count 2 interval 1 second
   or
   RP/0/RSP0/CPU0:router(config-sla-prof-pb)# send packet every 100 milliseconds | Configures the number and timing of packets sent by a probe in an operations profile.                                              |
| Step 3 | **packet size** **bytes** *(test pattern (hex 0xHHHHHHHH | pseudo-random))*          | (CFM loopback probe types only) Configures the minimum size (in bytes) for outgoing probe packets, including padding when necessary. Use the test pattern keyword to specify a hexadecimal string to use as the padding characters, or a pseudo-random bit sequence. The default padding is 0’s. |
|        | **Example:** RP/0/RSP0/CPU0:router(config-sla-prof-pb)# packet size 9000          |                                                                                                                                        |
Configuring SLA Statistics Measurement in a Profile

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

Prerequisites

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

To configure one-way delay measurements, be sure that the following clocking prerequisites are met:

- Frequency synchronization is configured globally in its default line timing mode (The clock-interface timing mode command is not configured.)
- The clock interface (Sync 0 / Sync 1) on the RSP is configured as a DTI port using the port-parameters dti command.
- Valid DTI input signals are available for the RSP clock-interface ports.
- Both Local and remote routers are using DTI timing input signals.

For more information about frequency synchronization configuration, see the “Configuring Ethernet Interfaces on the Cisco ASR 9000 Series Router”.

Restrictions

One-way delay and jitter measurements are not supported by cfm-loopback profile types.
To configure SLA statistics measurement in a profile, perform the following steps beginning in SLA profile configuration mode:

**SUMMARY STEPS**

1. `statistics measure {one-way-delay-ds | one-way-delay-sd | one-way-jitter-ds | one-way-jitter-sd | round-trip-delay | round-trip-jitter}
2. `aggregate {bins count width width | none}
3. `buckets size number {per-probe | probes}
4. `buckets archive number
5. `end
   or
   `commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**
- `statistics measure {one-way-delay-ds | one-way-delay-sd | one-way-jitter-ds | one-way-jitter-sd | round-trip-delay | round-trip-jitter}`
  
  **Example:**
  RP/0/RSP0/CPU0:router(config-sla-prof) #
  `statistics measure round-trip-delay`
  |
  Enables the collection of SLA statistics, and enters SLA profile statistics configuration mode.
| **Step 2**
- `aggregate {bins count width width | none}
  
  **Example:**
  RP/0/RSP0/CPU0:router(config-sla-prof-stat-cfg) #
  `- aggregate bins 100 width 10000`
  |
  Configures the size and number of bins into which to aggregate the results of statistics collection.
| **Step 3**
- `buckets size number {per-probe | probes}
  
  **Example:**
  RP/0/RSP0/CPU0:router(config-sla-prof-stat-cfg) #
  `- buckets size 100 per-probe`
  |
  Configures the size of the buckets in which statistics are collected.
How to Configure Ethernet OAM

Configuring a Schedule for an SLA Operation Probe in a Profile

This section describes how to configure a schedule for an SLA operation probe on an ongoing basis within an SLA profile. For information about how to configure a schedule for a limited, on-demand SLA operation, see the “Configuring an On-Demand SLA Operation” section on page 139.

To configure a schedule for an SLA operation probe, perform the following steps beginning in SLA profile configuration mode:

**SUMMARY STEPS**

1. `schedule every week [day] [at hh:mm] [for duration [seconds | minutes | hours | days | week]]`
   or
   `schedule every day [at hh:mm] [for duration [seconds | minutes | hours | days | week]]`
   or
   `schedule every number [hours | minutes] [at hh:mm] [for duration [seconds | minutes | hours | days | week]]`

2. `end`
   or
   `commit`

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 4</strong></td>
<td></td>
</tr>
<tr>
<td><code>buckets archive number</code></td>
<td>Configures the number of buckets to store in memory.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-sla-prof-stat-cfg)</td>
<td># buckets archive 50</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td></td>
</tr>
<tr>
<td><code>end</code> or <code>commit</code></td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-sla-prof-stat-cfg)</td>
<td># commit</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

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.
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| **Step 1**  
 schedule every week on day [at hh:mm] [for duration {seconds | minutes | hours | days | week}]  
 or  
 schedule every day [at hh:mm] [for duration {seconds | minutes | hours | days | week}]  
 or  
 schedule every number {hours | minutes} [first at hh:mm[.ss]] [for duration {seconds | minutes | hours | days | week}]  
  
*Example:*  
RP/0/RSP0/CPU0:router(config-sla-prof)#  
schedule every week on Monday at 23:30 for 1 hour  
or  
RP/0/RSP0/CPU0:router(config-sla-prof)#  
schedule every day at 11:30 for 5 minutes  
or  
RP/0/RSP0/CPU0:router(config-sla-prof)#  
schedule every 2 hours first at 13:45:01  
or  
RP/0/RSP0/CPU0:router(config-sla-prof)#  
schedule every 6 hours for 2 hours | Schedules an operation probe in a profile. A profile may contain only one schedule. |
| **Step 2**  
 end  
or  
 commit  
  
*Example:*  
RP/0/RSP0/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 an SLA Operation

This section describes how to configure an ongoing SLA operation on a MEP using an SLA profile.

SUMMARY STEPS

1. `interface [GigabitEthernet | TenGigE] interface-path-id`
2. `ethernet cfm`
3. `mep domain domain-name service service-name mep-id id-number`
4. `sla operation profile profile-name target {mep-id id | mac-address mac-address}`
5. 
6. `end`
   or
   `commit`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>`interface [GigabitEthernet</td>
<td>TenGigE] interface-path-id`</td>
</tr>
<tr>
<td>Note Use the <code>show interfaces</code> command to see a list of all interfaces currently configured on the router.</td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if)# interface gigabitethernet 0/1/0/1</td>
<td>For more information about the syntax for the router, use the question mark (?) online help function.</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>ethernet cfm</code></td>
<td>Enters interface CFM configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if)# ethernet cfm</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><code>mep domain domain-name service service-name mep-id id-number</code></td>
<td>Creates a MEP on an interface and enters interface CFM MEP configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if-cfm)# mep domain Dm1 service Sv1 mep-id 1</td>
<td></td>
</tr>
</tbody>
</table>
Configuring an On-Demand SLA Operation

Beginning in Cisco IOS XR Release 4.0, the Cisco ASR 9000 Series Router supports configuration of on-demand SLA operations to run on an as-needed basis for a finite period of time.

This section includes the following topics:

- Configuration Guidelines, page 139
- Configuring an On-Demand Ethernet SLA Operation for CFM Delay Measurement, page 140
- Configuring an On-Demand Ethernet SLA Operation for CFM Loopback, page 141

Configuration Guidelines

When you configure on-demand SLA operations, consider the following guidelines:

- Each MEP supports up to 50 on-demand operations.
- Each card supports up to 250 on-demand operations.
- On-demand Ethernet SLA operations can be run in addition to any other ongoing scheduled SLA operations that you might have configured, and use similar amounts of CPU and router memory. When configuring an on-demand Ethernet SLA operation, you should consider your existing SLA operation configuration and the potential impact of additional packet processing to your normal operations.

### Step 4

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>`sla operation profile profile-name target {mep-id id</td>
<td>mac-address mac-address}`</td>
</tr>
</tbody>
</table>

**Example:**

RP/0/RSP0/CPU0:router(config-if-cfm-mep)# sla operation profile Profile_1 target mac-address 01:23:45:67:89:ab

### Step 5

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>end</code> or <code>commit</code></td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>

**Example:**

RP/0/RSP0/CPU0:router(config-sla-prof-stat-cfg)# commit

- When you issue the `end` command, the system prompts you to commit changes:
  
  Uncommitted changes found, commit them before exiting(yes/no/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.

---

**Command or Action Purpose**

<table>
<thead>
<tr>
<th>Step 4</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>**sla operation profile profile-name target {mep-id id</td>
<td>mac-address mac-address}**</td>
<td>Creates an operation instance from a MEP to a specified destination.</td>
</tr>
</tbody>
</table>

**Example:**

RP/0/RSP0/CPU0:router(config-if-cfm-mep)# sla operation profile Profile_1 target mac-address 01:23:45:67:89:ab

<table>
<thead>
<tr>
<th>Step 5</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>end</strong> or <strong>commit</strong></td>
<td>Saves configuration changes.</td>
<td></td>
</tr>
</tbody>
</table>

**Example:**

RP/0/RSP0/CPU0:router(config-sla-prof-stat-cfg)# commit

- When you issue the `end` command, the system prompts you to commit changes:
  
  Uncommitted changes found, commit them before exiting(yes/no/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.
If you do not specify a schedule for the on-demand operation, the probe defaults to running one time beginning two seconds from the execution of the command, and runs for a ten-second duration.

If you do not specify the statistics for the probe to measure, it defaults to measuring all statistics, including the following statistics by probe type:
- CFM loopback—Two-way delay and jitter is measured by default.
- CFM delay measurement—One-way delay and jitter in both directions, in addition to two-way delay and jitter is measured by default.

The default operation mode is synchronous, where progress of the operation is reported to the console and the output of the statistics collection is displayed.

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

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| `ethernet sla on-demand operation type cfm-delay-measurement probe [priority number] [send {packet {once | every number {milliseconds | seconds | minutes | hours}} | burst {once | every number {seconds | minutes | hours}}] [packet count number interval number {milliseconds | seconds}] domain domain-name source interface type interface-path-id target {mac-address H.H.H.H | mep-id id-number} [statistics measure {one-way-delay-ds | one-way-delay-sd | one-way-jitter-ds | one-way-jitter-sd | round-trip-delay | round-trip-jitter}] [aggregate {none | bins number width milliseconds}] [buckets {archive number | size number {per-probe | probes}}] [schedule {now | at hh:mm:ss} [day [month [year]]] | in number {seconds | minutes | hours} [for duration {seconds | minutes | hours}] [repeat every number {seconds | minutes | hours} count probes]] [asynchronous]` | Configures an on-demand Ethernet SLA operation for CFM delay measurement. The example shows a minimum configuration, that specifies the local domain and source interface and target MEP, using the following defaults:
- Send a burst once for a packet count of 10 and interval of 1 second (10-second probe).
- Use default class of service (CoS) for the egress interface.
- Measure all statistics, including both one-way and round-trip delay and jitter statistics.
- Aggregate statistics into one bin.
- Schedule now.
- Display results on the console. |

Example:

```
RP/0/RSP0/CPU0:router# ethernet sla on-demand operation type cfm-delay-measurement probe domain D1 source interface TenGigE 0/6/1/0 target mep-id 100
```
Configuring an On-Demand Ethernet SLA Operation for CFM Loopback

To configure an on-demand Ethernet SLA operation for CFM loopback, use the following command in privileged EXEC configuration mode:

```
ethernet sla on-demand operation type cfm-loopback probe
[packet size bytes [test pattern {hex 0xHHH | pseudo-random}]]
[priority number]
[send {packet {once | every number {milliseconds | seconds | minutes | hours}} | burst {once | every number {seconds | minutes | hours}}] packet count number interval number [milliseconds | seconds]]
domain domain-name source interface type interface-path-id target
{mac-address H.H.H.H | mep-id id-number}
[statistics measure {round-trip-delay | round-trip-jitter}][aggregate
{none | bins number width milliseconds}][buckets {archive number | size
number {per-probe | probes}}]
[schedule {now | at hh:mm:ss [day [month [year]]]} | in number {seconds |
minutes | hours}][for duration {seconds | minutes | hours}][repeat every
number {seconds | minutes | hours} count probes][asynchronous]
```

Example:
```
RP/0/RSP0/CPU0:router# ethernet sla on-demand operation type
  cfm-loopback probe packet size 1500 domain D1 source interface
  TenGigE 0/6/1/0 target mep-id 100
```

Verifying SLA Configuration

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

```
show ethernet sla configuration-errors [domain domain-name] [interface
  interface-path-id] [profile profile-name]
show ethernet sla operations [detail] [domain domain-name] [interface
  interface-path-id] [profile profile-name]
```

Configuring Ethernet LMI

To configure Ethernet LMI, complete the following tasks:

- Prerequisites for Configuring E-LMI, page 142
- Restrictions for Configuring E-LMI, page 142
- Creating EVCs for E-LMI, page 142 (required)
Configuring Ethernet OAM on the Cisco ASR 9000 Series Router

How to Configure Ethernet OAM

Prerequisites for Configuring E-LMI

Before you configure E-LMI on the Cisco ASR 9000 Series Router, be sure that you complete the following requirements:

- Identify the local and remote UNIs in your network where you want to run E-LMI, and define a naming convention for them.
- Enable E-LMI on the corresponding CE interface link on a device that supports E-LMI CE operation, such as the Cisco Catalyst 3750 Metro Series Switches.

Restrictions for Configuring E-LMI

When configuring E-LMI, consider the following restrictions:

- E-LMI is not supported on subinterfaces or bundle interfaces. E-LMI is configurable on Ethernet physical interfaces only.

Creating EVCs for E-LMI

EVCs for E-LMI on the Cisco ASR 9000 Series Router are established by first configuring EFPs (Layer 2 subinterfaces) on the local UNI physical Ethernet interface link to the CE where E-LMI will be running, and also on the remote UNI link. Then, the EFPs need to be assigned to an L2VPN bridge domain to create the EVC.

To create EVCs, complete the following tasks:

- Configuring EFPs, page 142 (required)
- Configuring a Bridge Group and Assigning EFPs to a Bridge Domain, page 145 (required)

Configuring EFPs

This section describes the basic configuration of an EFP. For more information about configuration of other supported Layer 2 services, see the Cisco ASR 9000 Series Aggregation Services Routers L2VPN and Ethernet Services Configuration Guide.
To configure an EFP, complete the following tasks:

**SUMMARY STEPS**

1. `configure`
2. `interface [GigabitEthernet | TenGigE] interface-path-id.subinterface l2transport`
3. `encapsulation dot1q vlan-id [, untagged | , vlan-id | –vlan-id] [exact | ingress source-mac mac-address | second-dot1q vlan-id]`
4. `end`
   or
   `commit`
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface [GigabitEthernet</td>
<td>TenGigE] interface-path-id.subinterface l2transport</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/0/0/0.0 l2transport</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> encapsulation dot1q vlan-id [, untagged</td>
<td>, vlan-id</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-subif)# encapsulation dot1q 1-20</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-subif)# commit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• When you issue the <strong>end</strong> command, the system prompts you to commit changes: Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>– Entering <strong>yes</strong> saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>– Entering <strong>no</strong> exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>– Entering <strong>cancel</strong> leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Use the <strong>commit</strong> command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>
Configuring a Bridge Group and Assigning EFPs to a Bridge Domain

To configure a bridge group and assign EFPs to a bridge domain to create an EVC, complete the following steps:

**SUMMARY STEPS**

1. configure
2. l2vpn
3. bridge group name
4. bridge-domain name
5. interface {GigabitEthernet | TenGigE} interface-path-id.subinterface
6. end
   or
   commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> l2vpn</td>
<td>Enters L2VPN configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config)# l2vpn</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> bridge group bridge-group-name</td>
<td>Creates a bridge group and enters L2VPN bridge group configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group BG1</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> bridge-domain bridge-domain-name</td>
<td>Creates a bridge domain and enters L2VPN bridge group bridge domain configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-l2vpn-bg)# bridge-domain BD1</td>
<td></td>
</tr>
</tbody>
</table>
How to Configure Ethernet OAM

Configuring Ethernet OAM on the Cisco ASR 9000 Series Router

How to Configure Ethernet OAM

HC-146
Cisco ASR 9000 Series Aggregation Services Router Interface and Hardware Component Configuration Guide
OL-26061-03

Configuring Ethernet CFM for E-LMI

The Cisco ASR 9000 Series Router uses Ethernet CFM to monitor EVC status for E-LMI. To use CFM for E-LMI, a CFM maintenance domain and service must be configured on the router and the EFPs must be configured as CFM Up MEPs.

To configure Ethernet CFM for E-LMI, complete the following tasks:

- Configuring Ethernet CFM, page 108 (required)
- Configuring EFPs as CFM Up MEPs, page 147 (required)

Configuring Ethernet CFM

The minimum configuration to support E-LMI using Ethernet CFM is to configure a CFM maintenance domain and service on the router. Other CFM options can also be configured.

For more information about the tasks to configure Ethernet CFM, see the “Configuring Ethernet CFM” section on page 108.
Configuring EFPs as CFM Up MEPs

This section describes the minimum tasks required to configure EFPs as CFM MEPs. For more information about configuring CFM MEPs, see the “Configuring CFM MEPs” section on page 120.

To configure EFPs as CFM MEPs, complete the following tasks for each E-LMI EFP:

**SUMMARY STEPS**

1. configure
2. interface {GigabitEthernet | TenGigE} interface-path-id.subinterface
3. ethernet cfm
4. mep domain domain-name service service-name mep-id id-number
5. end or commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface gigabitethernet interface-path-id.subinterface 12transport</td>
<td>Enters Layer 2 subinterface configuration mode for the EFP.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/0/0.0 12transport</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ethernet cfm</td>
<td>Enters Ethernet CFM interface configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-subif)# ethernet cfm</td>
<td></td>
</tr>
</tbody>
</table>
Configuring UNI Names on the Physical Interface

It is recommended that you configure UNI names on the physical interface links to both the local and remote UNIs to aid in management for the E-LMI protocol. To configure UNI names, complete the following tasks on the physical interface links to both the local and remote UNIs:

**SUMMARY STEPS**

1. configure
2. interface [GigabitEthernet | TenGigE] interface-path-id
3. ethernet uni id name
4. end
   or
   commit

---

**Command or Action**

**Purpose**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4 mep domain domain-name service service-name mep-id id-number</td>
<td>Creates a MEP on an interface and enters interface CFM MEP configuration mode.</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-if-cfm)# mep domain GLOBAL service CustomerA mep-id 22</td>
<td></td>
</tr>
<tr>
<td>Step 5 end</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>or commit</td>
<td>- When you issue the end command, the system prompts you to commit changes:</td>
</tr>
<tr>
<td>Example: RP/0/RSP0/CPU0:router(config-if-cfm-mep)# commit</td>
<td>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:</td>
</tr>
<tr>
<td></td>
<td>- Entering yes saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.</td>
</tr>
<tr>
<td></td>
<td>- Entering no exits the configuration session and returns the router to EXEC mode without committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>- Entering cancel leaves the router in the current configuration session without exiting or committing the configuration changes.</td>
</tr>
<tr>
<td></td>
<td>• Use the commit command to save the configuration changes to the running configuration file and remain within the configuration session.</td>
</tr>
</tbody>
</table>
DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>configure</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Enters interface configuration mode for the physical interface.</td>
</tr>
<tr>
<td>interface [GigabitEthernet</td>
<td>TenGigE] interface-path-id</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/0/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td>Specifies a name (up to 64 characters) for the Ethernet UNI interface link.</td>
</tr>
<tr>
<td>ethernet uni id name</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>RP/0/RSP0/CPU0:router(config-if)# ethernet uni id PE1-CustA-Slot0-Port0</td>
<td></td>
</tr>
</tbody>
</table>
| **Step 4**        | Saves configuration changes.  
| end or commit      |                                                   |
| Example:           |                                                   |
| RP/0/RSP0/CPU0:router(config-if)# commit |                                                   |

- **Enabling E-LMI on the Physical Interface**

  The Cisco ASR 9000 Series Router supports the E-LMI protocol only on physical Ethernet interfaces. To enable E-LMI, complete the following tasks on the physical Ethernet interface link to the local UNI:

  **SUMMARY STEPS**

  1. configure
  2. interface [GigabitEthernet | TenGigE] interface-path-id
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface [GigabitEthernet</td>
<td>TenGigE] interface-path-id</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router# interface gigabitethernet 0/0/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> ethernet lmi</td>
<td>Enables Ethernet Local Management Interface operation on an interface and enters interface Ethernet LMI configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if)# ethernet lmi</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> end or commit</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td><strong>Example:</strong> RP/0/RSP0/CPU0:router(config-if-lmi)# commit</td>
<td></td>
</tr>
</tbody>
</table>

- 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 the Polling Verification Timer

The MEF T392 Polling Verification Timer (PVT) specifies the allowable time between transmission of a STATUS message and receipt of a STATUS ENQUIRY from the UNI-C before recording an error. The default value is 15 seconds.

To modify the default value or disable the PVT altogether, complete the following tasks:

**SUMMARY STEPS**

1. configure
2. interface [GigabitEthernet | TenGigE] interface-path-id
3. ethernet lmi
4. polling-verification-timer {interval | disable}
5. end
   or
   commit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

RP/0/RSP0/CPU0:router# configure

| **Step 2** interface [GigabitEthernet | TenGigE] interface-path-id | Enters interface configuration mode for the physical interface. |

**Example:**

RP/0/RSP0/CPU0:router# interface gigabitethernet 0/0/0/0

| **Step 3** ethernet lmi         | Enables Ethernet Local Management Interface operation on an interface and enters interface Ethernet LMI configuration mode. |

**Example:**

RP/0/RSP0/CPU0:router(config-if)# ethernet lmi
Configuring Ethernet OAM

How to Configure Ethernet OAM

Configuring the Status Counter

The MEF N393 Status Counter value is used to determine E-LMI operational status by tracking receipt of consecutive good packets or successive expiration of the PVT on packets. The default counter is four, which means that while the E-LMI protocol is in Down state, four good packets must be received consecutively to change the protocol state to Up, or while the E-LMI protocol is in Up state, four consecutive PVT expirations must occur before the state of the E-LMI protocol is changed to Down on the interface.

To modify the status counter default value, complete the following tasks:

**SUMMARY STEPS**

1. configure
2. interface [GigabitEthernet | TenGigE] interface-path-id
3. ethernet lmi
4. status-counter threshold
5. end
   or
   commit

---

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 4 polling-verification-timer {interval</td>
<td>disable}</td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if-lmi)# polling-verification-timer 30</td>
</tr>
<tr>
<td>Step 5 end</td>
<td>Saves configuration changes.</td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>commit</td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td>RP/0/RSP0/CPU0:router(config-if-lmi)# commit</td>
</tr>
</tbody>
</table>
## DETAILED STEPS

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td><code>configure</code></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
<td></td>
</tr>
<tr>
<td>Step 2</td>
<td>`interface [GigabitEthernet</td>
<td>TenGigE] interface-path-id`</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# interface gigabitethernet 0/0/0</td>
<td></td>
</tr>
<tr>
<td>Step 3</td>
<td><code>ethernet lmi</code></td>
<td>Enables Ethernet Local Management Interface operation on an interface and enters interface Ethernet LMI configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-if)# ethernet lmi</td>
<td></td>
</tr>
<tr>
<td>Step 4</td>
<td><code>status-counter threshold</code></td>
<td>Sets the MEF N393 Status Counter value that is used to determine E-LMI operational status by tracking receipt of consecutive good and bad packets from a peer. The default is 4.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-if-lmi)# status-counter 5</td>
<td></td>
</tr>
<tr>
<td>Step 5</td>
<td><code>end</code> or <code>commit</code></td>
<td>Saves configuration changes.</td>
</tr>
</tbody>
</table>
| **Example:** | RP/0/RSP0/CPU0:router(config-if-lmi)# commit | - 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. |
Disabling Syslog Messages for E-LMI Errors or Events

The E-LMI protocol tracks certain errors and events whose counts can be displayed using the `show ethernet lmi interfaces` command.

To disable syslog messages for E-LMI errors or events, complete the following tasks:

**SUMMARY STEPS**

1. `configure`
2. `interface [GigabitEthernet | TenGigE] interface-path-id`
3. `ethernet lmi`
4. `log {errors | events} disable`
5. `end` or `commit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><code>configure</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>`interface [GigabitEthernet</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# interface gigabitethernet 0/0/0/0</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><code>ethernet lmi</code></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router(config-if)# ethernet lmi</td>
</tr>
</tbody>
</table>
Disabling Use of the Cisco-Proprietary Remote UNI Details Information Element

To provide additional information within the E-LMI protocol, the Cisco IOS XR software implements a Cisco-proprietary information element called Remote UNI Details to send information to the CE about remote UNI names and states. This information element implements what is currently an unused identifier from the E-LMI MEF 16 specification.

To disable use of the Remote UNI Details information element, complete the following tasks:

SUMMARY STEPS

1. configure
2. interface [GigabitEthernet | TenGigE] interface-path-id
3. ethernet lmi
4. extension remote-uni disable
5. end
   or
   commit

Command or Action | Purpose
--- | ---
log {errors | events} disable | Turns off syslog messages for E-LMI errors or events.
end | 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.
### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td><strong>configure</strong>&lt;br&gt;Enters global configuration mode.</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>RP/0/RSP0/CPU0:router# configure</td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>**interface {GigabitEthernet</td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>ethernet lmi</strong>&lt;br&gt;Enables Ethernet Local Management Interface operation on an interface and enters interface Ethernet LMI configuration mode.&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router(config-if)# ethernet lmi</td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>extension remote-uni disable</strong>&lt;br&gt;Disables transmission of the Cisco-proprietary Remote UNI Details information element in E-LMI STATUS messages.&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router(config-if-lmi)# extension remote-uni disable</td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>end</strong>&lt;br&gt;or&lt;br&gt;<strong>commit</strong>&lt;br&gt;Saves configuration changes.&lt;br&gt;<strong>Example:</strong>&lt;br&gt;RP/0/RSP0/CPU0:router(config-if-lmi)# commit</td>
</tr>
</tbody>
</table>

- When you issue the **end** command, the system prompts you to commit changes:<br>Uncommitted changes found, commit them before exiting(yes/no/cancel)? [cancel]:
  - Entering **yes** saves configuration changes to the running configuration file, exits the configuration session, and returns the router to EXEC mode.
  - Entering **no** exits the configuration session and returns the router to EXEC mode without committing the configuration changes.
  - Entering **cancel** leaves the router in the current configuration session without exiting or committing the configuration changes.

- Use the **commit** command to save the configuration changes to the running configuration file and remain within the configuration session.
Verifying the Ethernet LMI Configuration

Use the `show ethernet lmi interfaces detail` command to display the values for the Ethernet LMI configuration for a particular interface, or for all interfaces. The following example shows sample output for the command:

```
RP/0/RSP0/CPU0:router# show ethernet lmi interfaces detail
Interface: GigabitEthernet0/0/0/0
  Ether LMI Link Status: Up
  UNI Id: PE1-CustA-Slot0-Port0
  Line Protocol State: Up
  MTU: 1514 (1 PDU reqd. for full report)
  CE-VLAN/EVC Map Type: Bundling (1 EVC)
  Configuration: Status counter 4, Polling Verification Timer 15 seconds
  Last Data Instance Sent: 0
  Last Sequence Numbers: Sent 0, Received 0

  Reliability Errors:
    Status Enq Timeouts: 0
    Invalid Report Type: 0

  Protocol Errors:
    Malformed PDUs: 0
    Invalid Message Type: 0
    Invalid Protocol Version: 0
    Invalid Sequence Number: 0
    Invalid Report Type: 0
    Invalid Mandatory IE: 0
    Unrecognized IE: 0

    Full Status Enq Received: never
    Full Status Sent: never
    PDU Received: never
    PDU Sent: never
    LMI Link Status Changed: 00:00:03 ago
    Last Protocol Error: never
    Counters cleared: never

  Sub-interface: GigabitEthernet0/0/0/0.0
    VLANs: 1-20
    EVC Status: Active
    EVC Type: Point-to-Point
    OAM Protocol: CFM
    CFM Domain: Global (level 5)
    CFM Service: CustomerA
    Remote UNI Count: Configured = 1, Active = 1

  Remote UNI Id
    PE1-CustA-Slot0-Port1 Status
    -------------- ------
    PE1-CustA-Slot0-Port1 Up
```

Troubleshooting Tips for E-LMI Configuration

This section describes some basic information for troubleshooting your E-LMI configuration in the following topics:

- Ethernet LMI Link Status Troubleshooting, page 158
- Ethernet LMI Line Protocol State Troubleshooting, page 158
- Ethernet LMI Error Counter Troubleshooting, page 159
- Ethernet LMI Remote UNI Troubleshooting, page 159
Ethernet LMI Link Status Troubleshooting

The E-LMI protocol operational status is reported in the “Ether LMI Link Status” or “ELMI state” fields in the output of forms of the `show ethernet lmi interfaces` command. To investigate a link status other than “Up,” consider the following guidelines:

- **Unknown (PVT disabled)**—Indicates that the Polling Verification Timer has been configured as disabled, so no status information can be provided. To see an “Up” or “Down” status, you must enable the PVT. For more information, see the “Configuring the Polling Verification Timer” section on page 151.

- **Down**—The E-LMI link status can be Down for the following reasons:
  - The PVT has timed out the number of times specified by the `status-counter` command. This indicates that STATUS ENQUIRY messages have not been received from the CE device. This can be for the following reasons:
    - The CE device is not connected to the PE device. Check that the CE device is connected to the interface on which E-LMI is enabled on the PE device.
    - The CE device is not sending Status Enquiries. Check that E-LMI is enabled on the CE interface which is connected to the PE device.
    - Protocol errors are causing the PVT to expire. The PVT is only reset when a valid (unerrored) STATUS ENQUIRY message is received.
  - The Line Protocol State is “Down” or “Admin Down.”
  - The protocol has not yet started on the interface because it does not have useful information to provide, such as the UNI Id or details about EVCs. This is a symptom of provisioning misconfiguration.

**Note** If the protocol is started, then E-LMI still responds to STATUS ENQUIRY messages when it is in “Down” state.

Ethernet LMI Line Protocol State Troubleshooting

The E-LMI line protocol state is reported in the “Line Protocol State” or “LineP State” fields in the output of forms of the `show ethernet lmi interfaces` command. The line protocol state is the state of the E-LMI protocol on the physical interface.

To investigate a line protocol state other than Up, consider the following guidelines:

- **Admin-Down**—The interface is configured with the `shutdown` command. Use the `no shutdown` command to bring the interface up.

- **Down**—Indicates a fault on the interface. Run the `show interfaces` command to display both the interface state and the interface line protocol state for more information, and take the following actions to investigate further:
  - If both states are Down, this suggests a physical problem with the link (for example, the cable is not plugged into either the PE or CE device).
  - If the interface state is Up but the line protocol state is Down, this suggests that an OAM protocol has brought the line protocol state down due to a fault. Use the `show efd interface` command for more information.
Ethernet LMI Error Counter Troubleshooting

The `show ethernet lmi interfaces` command displays two sections of error counters:

- **Reliability Errors**—Can indicate that messages are being lost between the PE and CE devices. The timers in the last block of the output should indicate that messages are being sent and received by the PE device.

- **Protocol Errors**—Indicates that the CE device is sending packets to the PE device, but the PE does not understand those packets. This suggests an incorrect configuration of the E-LMI protocol on the CE side, or corruption of the packets on the path between the CE and PE. E-LMI packets have a strictly defined structure in the MEF 16 standard, and any deviation from that results in a protocol error. The PE will not respond to any packets that are malformed and result in a protocol error.

Immediately after configuring E-LMI, all of the error counters should be zero, with the possible exception of the Status Enq Timeouts counter. The Status Enq Timeouts counter can be non-zero if the E-LMI protocol was started on the PE interface before being started on the corresponding CE interface. However, once the protocol is started on both devices, this counter should stop increasing.

If the Status Enq Timeouts counter is non-zero and is increasing, this indicates that enquiries are not being received from the CE device. This can be due to the following conditions:

- The CE device is not connected or not sending STATUS ENQUIRY messages. For more information, see also the “Ethernet LMI Link Status Troubleshooting” section on page 158.

- The Polling Timer on the CE device is configured to a value greater than the PVT on the PE device. Verify that the value of the `polling-verification-timer` command on the PE device is larger than the value of the CE’s Polling Timer.

For more information, see also the documentation for the `show ethernet lmi interfaces` command in the Cisco ASR 9000 Aggregation Services Router Interfaces and Hardware Component Command Reference.

Ethernet LMI Remote UNI Troubleshooting

Information about the Remote UNIs is reported in the output of the `show ethernet lmi interfaces detail` command. The Remote UNI ID field displays the name of the UNI as configured by the `ethernet uni id` command, or it displays the CFM MEP ID of the UNI when the UNI name has not been configured.

If the Remote UNI is missing from the table altogether, this can be due to the following conditions:

- The remote UNI's EFP is missing from the bridge-domain in L2VPN configuration. Use the `show ethernet cfm configuration-errors` command to verify the configuration.

- A CFM MEP has not been configured on the remote UNI’s EFP.

Configuring UDLD

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

Perform these steps to configure UDLD protocol on an interface.

**SUMMARY STEPS**

1. `configure`
2. `interface [GigabitEthernet | TenGigE] interface-path-id`
3. `ethernet udld`
Configuring Ethernet OAM on the Cisco ASR 9000 Series Router

**How to Configure Ethernet OAM**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>configure</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

**Example:**

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

| Step 2 | interface (GigabitEthernet | TenGigE) interface-path-id | Enters interface configuration mode and specifies the Ethernet interface name and notation rack/slot/module/port. |
|--------|-----------------------------|-----------------------------|

**Example:**

```
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.

<table>
<thead>
<tr>
<th>Step 3</th>
<th>ethernet udld</th>
<th>Enables ethernet UDLD function and enters interface Ethernet UDLD configuration mode.</th>
</tr>
</thead>
</table>

**Example:**

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

| Step 4 | mode (normal | aggressive) | (Optional) Specifies the mode of operation for UDLD. The options are normal and aggressive. |
|--------|---------------|----------------|

**Example:**

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

<table>
<thead>
<tr>
<th>Step 5</th>
<th>message-time (7-90)</th>
<th>(Optional) Specifies the message time (in seconds) to use for the UDLD protocol. The value ranges between 7 to 90 seconds.</th>
</tr>
</thead>
</table>

**Example:**

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

<table>
<thead>
<tr>
<th>Step 6</th>
<th>logging disable</th>
<th>(Optional) This command suppresses the operational UDLD syslog messages.</th>
</tr>
</thead>
</table>

**Example:**

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

<table>
<thead>
<tr>
<th>Step 7</th>
<th>end</th>
<th>Ends the configuration session and exits to the EXEC mode.</th>
</tr>
</thead>
</table>

**Example:**

```
RP/0/RSP0/CPU0:router(config-if-udld)# end
```
Configuration Examples for Ethernet OAM

This section provides the following configuration examples:

- Configuration Examples for EOAM Interfaces, page 161
- Configuration Examples for Ethernet CFM, page 163
- Configuration Examples for Ethernet SLA, page 174
- Configuration Example for Ethernet LMI, page 182

Configuration Examples for EOAM Interfaces

This section provides the following configuration examples:

- Configuring an Ethernet OAM Profile Globally: Example, page 161
- Configuring Ethernet OAM Features on an Individual Interface: Example, page 162
- Configuring Ethernet OAM Features to Override the Profile on an Individual Interface: Example, page 162
- Configuring a Remote Loopback on an Ethernet OAM Peer: Example, page 163
- Clearing Ethernet OAM Statistics on an Interface: Example, page 163
- Enabling SNMP Server Traps on a Router: Example, page 163

Configuring an Ethernet OAM Profile Globally: Example

The following 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 low 10000000 high 60000000
    frame window 60
    frame threshold low 10000000 high 60000000
    frame-period window 60000
    frame-period threshold low 100 high 12000000
    frame-seconds window 900000
    frame-seconds threshold 3 threshold 900
  exit
mib-retrieval
  connection timeout 30
  require-remote mode active
  require-remote link-monitoring
  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

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

```bash
configure terminal
interface TenGigE 0/1/0/0
ethernet oam
link-monitor
  symbol-period window 60000
  symbol-period threshold low 10000000 high 60000000
  frame window 60
  frame threshold low 10000000 high 60000000
  frame-period window 60000
  frame-period threshold low 100 high 12000000
  frame-seconds window 900000
  frame-seconds threshold 3 threshold 900
exit
mib-retrieval
  connection timeout 30
  require-remote mode active
  require-remote link-monitoring
  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

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

```bash
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

Configuring a Remote Loopback on an Ethernet OAM Peer: Example

The following example shows how to configure a remote loopback on an Ethernet OAM peer:

```plaintext
configure terminal
interface gigabitethernet 0/1/5/6
ethernet oam
profile Profile_1
remote-loopback
```

The following example shows how to start a remote loopback on a configured Ethernet OAM interface:

```plaintext
ethernet oam loopback enable TenGigE 0/6/1/0
```

Clearing Ethernet OAM Statistics on an Interface: Example

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

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

Enabling SNMP Server Traps on a Router: Example

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

```plaintext
configure terminal
ethernet oam profile Profile_1
snmp-server traps ethernet oam events
```

Configuration Examples for Ethernet CFM

This section includes the following examples:

- Ethernet CFM Domain Configuration: Example, page 164
- Ethernet CFM Service Configuration: Example, page 164
- Flexible Tagging for an Ethernet CFM Service Configuration: Example, page 164
- Continuity Check for an Ethernet CFM Service Configuration: Example, page 164
- MIP Creation for an Ethernet CFM Service Configuration: Example, page 164
- Cross-check for an Ethernet CFM Service Configuration: Example, page 164
- Other Ethernet CFM Service Parameter Configuration: Example, page 165
- MEP Configuration: Example, page 165
- Ethernet CFM Show Command: Examples, page 165
- AIS for CFM Configuration: Examples, page 168
- AIS for CFM Show Commands: Examples, page 169
- EFD Configuration: Examples, page 173
- Displaying EFD Information: Examples, page 173
Configuring Ethernet OAM on the Cisco ASR 9000 Series Router

Configuration Examples for Ethernet OAM

Ethernet CFM Domain Configuration: Example

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

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

Ethernet CFM Service Configuration: Example

The following 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

The following example shows how to set the number of tags in CFM packets from up MEPs in a CFM domain service:

```
configure
eternet cfm
domain D1 level 1
  service S2 bridge group BG1 bridge-domain BD2
tag 1
commit
```

Continuity Check for an Ethernet CFM Service Configuration: Example

The following 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

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

```
mip auto-create all
commit
```

Cross-check for an Ethernet CFM Service Configuration: Example

The following 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

The following example shows how to configure other Ethernet CFM service options:

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

MEP Configuration: Example

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

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

Ethernet CFM Show Command: Examples

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

**Example 1**

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

```
RP/0/RSP0/CPU0:router# show ethernet cfm local maintenance-points
```

<table>
<thead>
<tr>
<th>Domain/Level</th>
<th>Service</th>
<th>Interface</th>
<th>Type</th>
<th>ID</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>fig/5</td>
<td>bay</td>
<td>Gi0/10/0/12.23456</td>
<td>Dn MEP</td>
<td>2</td>
<td>44:55:66</td>
</tr>
<tr>
<td>fig/5</td>
<td>bay</td>
<td>Gi0/0/1/0.1</td>
<td>MIP</td>
<td>55:66:77</td>
<td></td>
</tr>
<tr>
<td>fred/3</td>
<td>barney</td>
<td>Gi0/1/0/0.1</td>
<td>Up MEP</td>
<td>5</td>
<td>66:77:88</td>
</tr>
</tbody>
</table>

**Example 2**

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

```
RP/0/RSP0/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 GigabitEthernet0/1/2/3.234 and an Up MEP is also configured for domain blort, which is at the same level (5).
* A MEP is configured on interface GigabitEthernet0/3/2/1.1 for this domain/service, which has CC interval 100ms, but the lowest interval supported on that interface is 1s

**Example 3**

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

```
RP/0/RSP0/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)
Example 4

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

```
RP/0/RSP0/CPU0# 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
Up MEP on GigabitEthernet0/1/0/12.123, 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
R>     4 4455.6677.8899 Up    01:03:04       3456      0 234     0 3456
L     2 1122.3344.5566 Up    03:45        12345      0 300 12348
C     3 7788.9900.1122 Test  00:13        23456      6 20 2345
X     2 3344.5566.7788 Up    00:23        32546      0     0    3254
I     2 3344.5566.7788 Down  00:34        12345      0 300 12348
V     3 8899.0011.2233 Blocked 00:35        45678      0     0    4567
T     5 5566.7788.9900 Up    00:56        20345      0     0    2034
M     6 9900.1122.3344 Down  00:45        12345      0     0    1234
U>     7 6677.8899.0011 Up    00:02        45678      0     0    4567
```

Example 5

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

```
RP/0/RSP0/CPU0# show ethernet cfm peer meps detail
Domain dom3 (level 5), Service ser3
Down MEP on GigabitEthernet0/0/0/0, 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 GigabitEthernet0/0/0/0 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

Domain dom5 (level 2), Service ser5
Up MEP on Standby Bundle-Ether 1 MEP-ID 1

Peer MEP-ID 600, MAC 0001.0203.0401
CFM state: Ok (Standby), for 00:00:08, RDI received
Port state: Down
CCM defects detected: Defects below ignored on local standby MEP
I - Wrong Interval
R - Remote Defect received

CCMs received: 5
Out-of-sequence: 0
AIS for CFM Configuration: Examples

Example 1
The following example shows how to configure Alarm Indication Signal (AIS) transmission for a CFM domain service:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# ethernet cfm
RP/0/RSP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RSP0/CPU0:router(config-cfm-dmn)# service S1 bridge group BG1 bridge-domain BD2
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# ais transmission interval 1m cos 7
```

Example 2
The following 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/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# ethernet cfm
RP/0/RSP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RSP0/CPU0:router(config-cfm-dmn)# service S2 bridge group BG1 bridge-domain BD2
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# log ais
```
The following example shows how to configure AIS transmission on a CFM interface.

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# interface gigabitethernet 0/1/0/2
RP/0/RSP0/CPU0:router(config-if)# ethernet cfm
RP/0/RSP0/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, page 170
- show ethernet cfm local meps Command: Examples, page 170
show ethernet cfm interfaces ais Command: Example

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

```
RP/0/RSP0/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

<table>
<thead>
<tr>
<th>Interface (State)</th>
<th>Defects</th>
<th>Levels</th>
<th>Last started</th>
<th>Packets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gi0/1/0/0.234 (Up)</td>
<td>Dn 5 RPC</td>
<td>6</td>
<td>01:32:56 ago</td>
<td>5576</td>
</tr>
<tr>
<td>Gi0/1/0/0.567 (Up)</td>
<td>Up 0 M</td>
<td>2,3</td>
<td>00:16:23 ago</td>
<td>983</td>
</tr>
<tr>
<td>Gi0/1/0/1.1 (Dn)</td>
<td>Up D</td>
<td>7</td>
<td>01:02:44 ago</td>
<td>3764</td>
</tr>
<tr>
<td>Gi0/1/0/2 (Up)</td>
<td>Dn 0 RX</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

show ethernet cfm local meps Command: Examples

Example 1: Default

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

```
RP/0/RSP0/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 Gi1/1/0/1.234 (Up) Up 0/0 N A 7

Domain fred (level 5), Service barney
ID Interface (State) Dir MEPs/Err RD Defects AIS
----- ------------------------ --- -------- -- ------- ---
2 Gi0/1/0/0.234 (Up) Up 3/2 Y RPC 6
```

Example 2: Domain Service

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

```
RP/0/RSP0RSP0/CPU0:router# show ethernet cfm local meps domain foo service bar detail
```

```
Domain foo (level 6), Service bar
Up MEP on GigabitEthernet0/1/0/0.234, 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)
```
Example 3: Verbose

The following example shows how to display verbose statistics for MEPs in a domain service:

```
 RP/0/RSPORP0/CPU0# show ethernet cfm local meps domain foo service bar verbose
```

Note: The Discarded CCMs field is not displayed when the number is zero (0). It is unusual for the count of discarded CCMs to be anything other than zero, since CCMs are only discarded when the limit on the number of peer MEPs is reached.
## Configuration Examples for Ethernet OAM

<table>
<thead>
<tr>
<th>Packet</th>
<th>Sent</th>
<th>Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCN</td>
<td>12345</td>
<td>67890 (out of seq: 6, discarded: 10)</td>
</tr>
<tr>
<td>LBN</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>LBR</td>
<td>0</td>
<td>5 (out of seq: 0, with bad data: 0)</td>
</tr>
<tr>
<td>AIS</td>
<td>0</td>
<td>46910</td>
</tr>
<tr>
<td>LCK</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>
Example 4: Detail
The following example shows how to display detailed statistics for MEPs in a domain service:

```
RP/0/RSP0/CPU0:router# show ethernet cfm local meps detail
```

Domain foo (level 6), Service bar
Up MEP on GigabitEthernet0/1/0/0.234, 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
Up MEP on GigabitEthernet0/1/0/0.234, 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
```

EFD Configuration: Examples

The following example shows how to enable EFD:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# ethernet cfm
RP/0/RSP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RSP0/CPU0:router(config-cfm-dmn)# service S1 down-meps
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# efd
```

The following example shows how to enable EFD logging:

```
RP/0/RSP0/CPU0:router# configure
RP/0/RSP0/CPU0:router(config)# ethernet cfm
RP/0/RSP0/CPU0:router(config-cfm)# domain D1 level 1
RP/0/RSP0/CPU0:router(config-cfm-dmn)# service S1 down-meps
RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# log efd
```

Displaying EFD Information: Examples

The following examples show how to display information about EFD:

- `show efd interfaces Command: Example, page 174`
- `show ethernet cfm local meps detail Command: Example, page 174`
show efd interfaces Command: Example

The following example shows how to display all interfaces that are shut down in response to an EFD action:

```
RP/0/RSP0/CPU0:router# show efd interfaces

Server VLAN MA
============
Interface   Clients
------------
GigE0/0/0/0.0    CFM
```

show ethernet cfm local meps detail Command: Example

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

```
RP/0/RSP0/CPU0:router# show ethernet cfm local meps detail

Domain foo (level 6), Service bar
Up MEP on GigabitEthernet0/1/0/0.234, 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
Up MEP on GigabitEthernet0/1/0/0.234, 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`.

Configuration Examples for Ethernet SLA

This section includes the following examples:

- Ethernet SLA Profile Type Configuration: Examples, page 175
- Ethernet SLA Probe Configuration: Examples, page 175
- Profile Statistics Measurement Configuration: Examples, page 176
Ethernet SLA Profile Type Configuration: Examples

The following examples show how to configure the different profile types supported by Ethernet SLA.

Example 1
This example configures a profile named “Prof1” for CFM loopback measurements:

```
configure
ethernet sla
  profile  Prof1 type cfm-loopback
commit
```

Example 2
This example configures a profile named “Prof1” for CFM delay measurements. Setting this type allows you to configure the probe to measure additional one-way delay and jitter statistics:

```
configure
ethernet sla
  profile  Prof1 type cfm-delay-measurement
commit
```

Ethernet SLA Probe Configuration: Examples

The following examples show how to configure some of the packet options for an Ethernet CFM loopback probe.

Example 1
This example shows how to configure sending a group of 100 packets in 100 ms intervals and repeat that burst every 60 seconds. Packets are padded to a size of 9000 bytes as needed using a hexadecimal test pattern of “abcdabcd,” and with a class of service value of 7:

```
configure
ethernet sla
  profile  Prof1 type cfm-loopback
    probe
      send burst every 60 seconds packet count 100 interval 100 milliseconds
      packet size 9000 test pattern hex 0xabcdabcd
      priority 7
commit
```

Note
The total length of a burst (packet count multiplied by the interval) must not exceed 1 minute.
Example 2
This example has the same characteristics as the configuration in Example 1, but sends a single burst of 50 packets, one second apart:

```plaintext
configure
  ethernet sla
    profile Prof1 type cfm-loopback
    probe
      send burst once packet count 50 interval 1 second
      packet size 9000 test pattern hex 0xabcdabc
dr
      priority 7
    commit
```

Example 3
This example shows how to configure a continuous stream of packets at 100 ms intervals for the duration of the probe. Packets are padded to a size of 9000 bytes as needed using a pseudo-random test pattern, and with a class of service value of 7:

```plaintext
configure
  ethernet sla
    profile Prof1 type cfm-loopback
    probe
      send burst every 60 seconds packet count 600 interval 100 milliseconds
      packet size 9000 test pattern pseudo-random
      priority 7
    commit
```

Profile Statistics Measurement Configuration: Examples

The following examples show how to configure the different types of statistics measurement.

Example 1
This example shows the two available types of statistics that can be measured by a CFM loopback SLA profile type:

```plaintext
configure
  ethernet sla
    profile Prof1 type cfm-loopback
    statistics measure round-trip-delay
    statistics measure round-trip-jitter
  commit
```

Example 2
This example shows how to configure measurement of round-trip delay and one-way jitter (from destination to source) for a CFM delay measurement SLA profile type:

```plaintext
Note
The CFM delay measurement profile type supports measurement of all round-trip and one-way delay and jitter statistics.
```

```plaintext
configure
  ethernet sla
    profile Prof1 type cfm-delay-measurement
    statistics measure round-trip-delay
    statistics measure one-way-jitter-ds
    commit
```
Scheduled SLA Operation Probe Configuration: Examples

The following examples show how to configure different schedules for an SLA operation probe.

**Example 1**
This example shows how to configure a probe to run hourly for a specified duration:

```plaintext
configure
ethernet sla
  profile Prof1 type cfm-delay-measurement
  schedule every 1 hours for 15 minutes
commit
```

**Example 2**
This example shows how to configure a probe to run daily for a specified period of time:

```plaintext
configure
ethernet sla
  profile Prof1 type cfm-delay-measurement
  schedule every day at 11:30 for 5 minutes
commit
```

**Example 3**
This example shows how to configure a probe to run weekly beginning at a specified time and for a specified duration:

```plaintext
configure
ethernet sla
  profile Prof1 type cfm-delay-measurement
  schedule every week on Monday at 23:30 for 1 hour
commit
```

Ethernet SLA Operation Probe Scheduling and Aggregation Configuration: Example

**Figure 14** shows a more comprehensive example of how some of the probe scheduling and measurement configuration works using aggregation. The following configuration supports some of the concepts shown in the figure:

```plaintext
configure
ethernet sla profile Prof1 type cfm-loopback
  probe
    send packet every 60 seconds
    schedule every 6 hours for 2 hours
    statistics measure round-trip-delay
    aggregate bins 3 width 30
    buckets size 2 per-probe
    buckets archive 4
commit
```
**Figure 14**  SLA Probe Scheduling Operation With Bin Aggregation

This example schedules a probe with the following characteristics:

- Sends packets 60 seconds apart (for a 2-hour probe, this results in sending 120 individual packets).
- Probe runs every 6 hours for 2 hours duration.
- Collects data into 2 buckets for every probe, so each bucket covers 1 hour of the 2-hour probe duration.
- Aggregates statistics within the buckets into 3 bins each in the following ranges:
  - Bin 1 contains samples in the range 0 to < 30 ms.
  - Bin 2 contains samples in the range 30 ms to < 60 ms.
  - Bin 3 contains samples in the range 60 ms or greater (unbounded).
- The last 4 buckets are saved in memory.

**Ongoing Ethernet SLA Operation Configuration: Example**

The following example shows how to configure an ongoing Ethernet SLA operation on a MEP:

```plaintext
interface gigabitethernet 0/1/0/1
ethernet cfm
mep domain Dm1 service Sv1 mep-id 1
sla operation profile Profile_1 target mac-address 01:23:45:67:89:ab s
commit
end
```
On-Demand Ethernet SLA Operation Basic Configuration: Examples

The following examples show how to configure on-demand Ethernet SLA operations.

Example 1

The following example shows how to configure a basic on-demand Ethernet SLA operation for a CFM loopback probe that by default will measure round-trip delay and round-trip jitter for a one-time, 10-second operation to the target MEP:

```
RP/0/RSP0/CPU0:router# ethernet sla on-demand operation type cfm-loopback probe domain D1 source interface TenGigE 0/6/1/0 target mep-id 1
```

Example 2

The following example shows how to configure a basic on-demand Ethernet SLA operation for a CFM delay measurement probe that by default will measure one-way delay and jitter in both directions, as well as round-trip delay and round-trip jitter for a one-time, 10-second operation to the target MEP:

```
RP/0/RSP0/CPU0:router# ethernet sla on-demand operation type cfm-delay-measurement probe domain D1 source interface TenGigE 0/6/1/0 target mep-id 1
```

Ethernet SLA Show Commands: Examples

The following examples show how to display information about configured SLA operations:

```
show ethernet sla operations Command: Example 1
RP/0/RSP0/CPU0:router# show ethernet sla operations interface gigabitethernet 0/1/0/1.1

Interface GigabitEthernet0/1/0/1.1
Domain mydom Service myser to 00AB.CDEF.1234
-------------------------------------------------------------------
Profile 'business-gold'
Probe type CFM-delay-measurement:
    bursts sent every 1min, each of 20 packets sent every 100ms
    packets padded to 1500 bytes with zeroes
    packets use priority value of 7
Measures RTT: 5 bins 20ms wide; 2 buckets/probe; 75/100 archived
Measures Jitter (interval 1): 3 bins 40ms wide; 2 buckets/probe; 50 archived
Scheduled to run every Sunday at 4am for 2 hours:
    last run at 04:00 25/05/2008
```

```
show ethernet sla configuration-errors Command: Example 2
RP/0/RSP0/CPU0:router# show ethernet sla configuration-errors

Errors:
-------
Profile 'gold' is not defined but is used on Gi0/0/0/0.0
Profile 'red' defines a test-pattern, which is not supported by the type
```

The following examples show how to display the contents of buckets containing SLA metrics collected by probes:

```
show ethernet sla statistics current Command: Example 3
RP/0/RSP0/CPU0:router# show ethernet sla statistics current interface GigabitEthernet 0/0/0/0.0

Interface GigabitEthernet 0/0/0/0.0
Domain mydom Service myser to 00AB.CDEF.1234
```
Profile 'business-gold', packet type 'cfm-loopback'
Scheduled to run every Sunday at 4am for 2 hours

Round Trip Delay
~~~~~~~~~~~~~~~~
2 buckets per probe

Bucket started at 04:00 Sun 17 Feb 2008 lasting 1 hour:
   Pkts sent: 2342; Lost: 2 (0%); Corrupt: 0 (0%); Misordered: 0 (0%)
   Min: 13ms; Max: 154ms; Mean: 28ms; StdDev: 11ms

Round Trip Jitter
~~~~~~~~~~~~~~~~
2 buckets per probe

Bucket started at 04:00 Sun 17 Feb 2008 lasting 1 hour:
   Pkts sent: 2342; Lost: 2 (0%); Corrupt: 0 (0%); Misordered: 0 (0%)
   Min: -5ms; Max: 8ms; Mean: 0ms; StdDev: 3.6ms

Bucket started at 05:00 Sun 17 Feb 2008 lasting 1 hour:
   Pkts sent: 2342; Lost: 2 (0%); Corrupt: 0 (0%); Misordered: 0 (0%)
   Min: 0; Max: 4; Mean: 1.4; StdDev: 1

show ethernet sla statistics history detail Command: Example 4

RP/0/RSP0/CPU0:router# show ethernet sla history detail GigabitEthernet 0/0/0/0.0

Interface GigabitEthernet 0/0/0/0.0
Domain mydom Service myser to 00AB.CDEF.1234

Profile 'business-gold', packet type 'cfm-loopback'
Scheduled to run every Sunday at 4am for 2 hours

Round Trip Delay
~~~~~~~~~~~~~~~~
2 buckets per probe

Bucket started at 04:00 Sun 17 Feb 2008 lasting 1 hour:
   Pkts sent: 2342; Lost: 2 (0%); Corrupt: 0 (0%); Misordered: 0 (0%)
   Min: 13ms, occurred at 04:43:29 on Sun 22 Aug 2010 UTC
   Max: 154ms, occurred at 05:10:32 on Sun 22 Aug 2010 UTC
   Mean: 28ms; StdDev: 11ms

Results suspect as more than 10 seconds time drift detected
Results suspect as scheduling latency prevented some packets being sent

Samples:
   Time sent  Result  Notes
   ---------  -------  -------
   04:00:01.324  23ms
   04:00:01.425  36ms
   04:00:01.525 - Timed Out

Round Trip Jitter
~~~~~~~~~~~~~~~~
2 buckets per probe

Bucket started at 04:00 Sun 17 Feb 2008, lasting 1 hour:
   Pkts sent: 2342; Lost: 2 (0%); Corrupt: 0 (0%); Misordered: 0 (0%)
   Min: -5ms; Max: 10ms; Mean: 0ms; StdDev: 3.6ms

Samples:
Time sent       Result  Notes
------------  --------  ----------
04:00:01.324         -        
04:00:01.425      13ms     
04:00:01.525         -  Timed out
...

show ethernet sla statistics history detail on-demand: Example 5

The following example shows how to display statistics for all full buckets for on-demand operations in detail:

RP/0/RSP0/CPU0/router #show ethernet sla statistics history detail on-demand

Interface GigabitEthernet0/0/0/0.1
Domain mydom Service myser to 0123.4567.890A

============================================================================
On-demand operation ID #1, packet type 'cfm-delay-measurement'
Started at 15:38 on 06 July 2010 UTC, runs every 1 hour for 1 hour

Round Trip Delay

1 bucket per probe

Bucket started at 15:38 on Tue 06 Jul 2010 UTC, lasting 1 hour:
  Pkts sent: 1200; Lost: 4 (0%); Corrupt: 600 (50%); Misordered: 0 (0%)
  Min: 13ms, occurred at 15:43:29 on Tue 06 Jul 2010 UTC
  Max: 154ms, occurred at 16:15:34 on Tue 06 Jul 2010 UTC
  Mean: 28ms; StdDev: 11ms

  Bins:
  Range              Samples    Cum. Count      Mean
  ------------  ------------  ------------  --------
  0 - 20 ms     194 (16%)     194 (16%)      17ms
  20 - 40 ms    735 (61%)     929 (77%)      27ms
  40 - 60 ms    212 (18%)    1141 (95%)      45ms
  > 60 ms        55 (5%)    1196            70ms

Bucket started at 16:38 on Tue 01 Jul 2008 UTC, lasting 1 hour:
  Pkts sent: 3600; Lost: 12 (0%); Corrupt: 1800 (50%); Misordered: 0 (0%)
  Min: 19ms, occurred at 17:04:08 on Tue 06 Jul 2010 UTC
  Max: 70ms, occurred at 16:38:00 on Tue 06 Jul 2010 UTC
  Mean: 28ms; StdDev: 11ms

  Bins:
  Range              Samples    Cum. Count      Mean
  ------------  ------------  ------------  --------
  0 - 20 ms     194 (16%)     194 (16%)      19ms
  20 - 40 ms    735 (61%)     929 (77%)      27ms
  40 - 60 ms    212 (18%)    1141 (95%)      45ms
  > 60 ms        55 (5%)    1196            64ms
Configuration Example for Ethernet LMI

Figure 15 shows a basic E-LMI network environment with a local UNI defined on a Cisco ASR 9000 Series Router functioning as the PE using Gigabit Ethernet interface 0/0/0/0, and connectivity to a remote UNI over Gigabit Ethernet interface 0/0/0/1.

The following configuration provides a basic E-LMI configuration for the environment shown in Figure 15, for the Cisco ASR 9000 Series Router as the PE device on the local UNI with physical Gigabit Ethernet interfaces 0/0/0/0 and 0/0/0/1:

```
RP/0/RSP0/CPU0:router# configure
|
| Configure the Local UNI EFPs
|
| RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/0/0/0.0 l2transport
| RP/0/RSP0/CPU0:router(config-subif)# #encapsulation dot1q 1-20
| RP/0/RSP0/CPU0:router(config-subif)# exit
| RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/0/0/1.1 l2transport
| RP/0/RSP0/CPU0:router(config-subif)# #encapsulation dot1q 1-20
| RP/0/RSP0/CPU0:router(config-subif)# exit
|
| Create the EVC
|
| RP/0/RSP0/CPU0:router(config)# l2vpn
| RP/0/RSP0/CPU0:router(config-l2vpn)# bridge group BG1
| RP/0/RSP0/CPU0:router(config-l2vpn-bg)# interface GigabitEthernet0/0/0/0.0
| RP/0/RSP0/CPU0:router(config-l2vpn-bg)# interface GigabitEthernet0/0/0/1.1
| RP/0/RSP0/CPU0:router(config-l2vpn-bg)# exit
| RP/0/RSP0/CPU0:router(config-l2vpn-bg)# exit
| RP/0/RSP0/CPU0:router(config-l2vpn)# exit
|
| Configure Ethernet CFM
|
| RP/0/RSP0/CPU0:router(config)# ethernet cfm
| RP/0/RSP0/CPU0:router(config-cfm)# domain GLOBAL level 5
| RP/0/RSP0/CPU0:router(config-cfm-dmn)# service CustomerA bridge group BG1 bridge-domain BD1
| RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# continuity-check interval 100ms
| RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# mep crosscheck mep-id 22
| RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# mep crosscheck mep-id 11
| RP/0/RSP0/CPU0:router(config-cfm-dmn-svc)# exit
| RP/0/RSP0/CPU0:router(config-cfm-dmn)# exit
| RP/0/RSP0/CPU0:router(config-cfm)# exit
| RP/0/RSP0/CPU0:router(config)# interface GigabitEthernet0/0/0.0 l2transport
```
Where to Go Next

When you have configured an Ethernet interface, you can configure individual VLAN subinterfaces on that Ethernet interface.

For information about modifying Ethernet management interfaces for the shelf controller (SC), route processor (RP), and distributed RP, see the “Advanced Configuration and Modification of the Management Ethernet Interface on the Cisco ASR 9000 Series Router” module later in this document.

For information about IPv6 see the Implementing Access Lists and Prefix Lists on Cisco IOS XR Software module in the Cisco IOS XR IP Addresses and Services Configuration Guide.

Additional References

The following sections provide references related to implementing Gigabit and 10-Gigabit Ethernet interfaces.

Related Documents

<table>
<thead>
<tr>
<th>Related Topic</th>
<th>Document Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet L2VPN</td>
<td>Cisco ASR 9000 Series Aggregation Services Router L2VPN and Ethernet Services Configuration Guide</td>
</tr>
<tr>
<td>Cisco IOS XR master command reference</td>
<td>Cisco IOS XR Master Commands List</td>
</tr>
<tr>
<td>Cisco IOS XR interface configuration commands</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
<tr>
<td>Information about user groups and task IDs</td>
<td>Cisco IOS XR Interface and Hardware Component Command Reference</td>
</tr>
</tbody>
</table>
## Standards

<table>
<thead>
<tr>
<th>Standards</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 802.1ag</td>
<td>Connectivity Fault Management</td>
</tr>
<tr>
<td>ITU-T Y.1731</td>
<td>OAM Functions and Mechanisms for Ethernet Based Networks</td>
</tr>
</tbody>
</table>

## MIBs

<table>
<thead>
<tr>
<th>MIBs</th>
<th>MIBs Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE8021-CFM-MIB</td>
<td><a href="http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">To locate and download MIBs for selected platforms using Cisco IOS XR Software, use the Cisco MIB Locator found at the following URL:</a></td>
</tr>
</tbody>
</table>

## RFCs

<table>
<thead>
<tr>
<th>RFCs</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

## Technical Assistance

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.</td>
<td><a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a></td>
</tr>
</tbody>
</table>